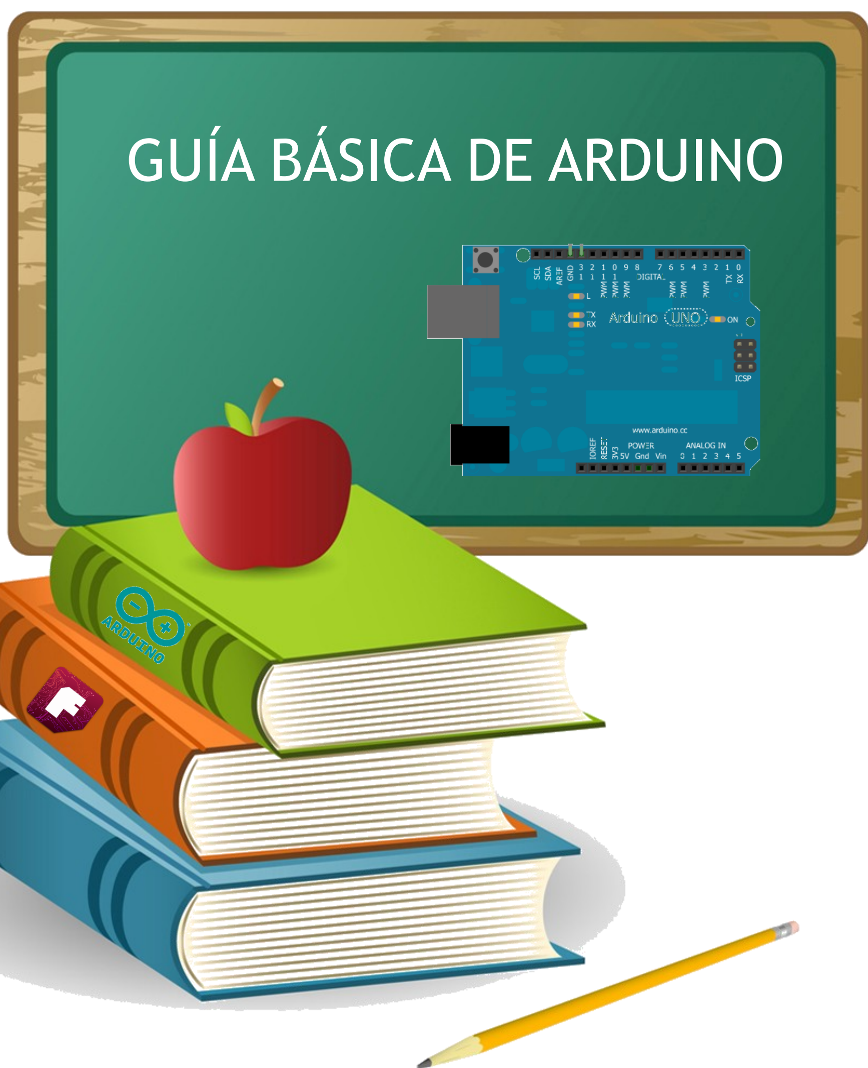
Una guía práctica sobre el mundo de Arduino



GUÍA BÁSICA DE ARDUINO

Hoja dejada en blanco de manera intencional

TABLA DE CONTENIDO

Hoja dejada en blanco de manera intencional

PROPÓSITO

Conocer el funcionamiento de las cosas es algo que nos hemos planteado desde el inicio de los tiempos; hoy en día nos enfrentamos a una realidad donde abundan la automatización, la domótica (automatización de las casas y edificios), la interacción de las personas con las máquinas, la electrónica, la mecánica y la programación.

Casi cualquier proceso que nos podamos imaginar tiene un porcentaje de dependencia de estas máquinas, por ejemplo: Tu despertador sonó a las 6am para que vinieras a la escuela o fueras al trabajo, esa máquina, reloj, trabajó durante toda la noche para al final avisarte que era hora de despertar.

El propósito de esta guía es abordar el concepto de computación física que es la capacidad de interacción y comunicación de una máquina con los humanos, usando sensores y actuadores. Las decisiones de esto las va a tomar un microcontrolador que se encuentra ubicado en la placa Arduino. La tarjeta **Arduino** es el corazón de la presente guía.

¿QUÉ VOY A APRENDER?

Muchas veces pensamos que los temas tecnológicos requieren de gran habilidad técnica y de un gran conocimiento, pero esto no es cierto. Queremos que con el desarrollo de esta guía entiendas que muchos de esos procesos tecnológicos son simples de entender y aquellos que son complejos son la unión de muchos procesos simples.

En esta guía vas a aprender a imaginar y aterrizar todas ideas a conceptos tangibles de los cuales te puedas sentir orgulloso, ya que fue tu idea y tu lo desarrollaste ;)

FUENTE DE INFORMACION

Wikipedia es una de las enciclopedia en la nube más grande que pueden existir, puedes encontrar gran variedad de información en distintos idiomas y eres libre de usarla para aprender.



La presente guía incorpora contenido de Wikipedia (texto e imágenes) con el animo de explicar los diversos conceptos que se enuncian. El contenido de la Wikipedia tomado en esta guía ha sido transcrito textualmente en algunos casos, en otros casos los conceptos se han reeditado para poder comprender más fácilmente la idea.

Para referenciar que hemos tomado contenido de Wikipedia, al lado de cada concepto técnico vas a encontrar el logo de Wikipedia de esta manera podrás leer más contenido si buscas ese mismo concepto en la Wikipedia.

Wikipedia es de contenido libre, de manera que todo el texto está disponible bajo la Licencia Creative Commons-Atribución-Compartir Igual 3.0(CC-BY-SA). La mayor parte del contenido también está disponible bajo la Licencia de Documentación Libre GNU (GFDL). Esto significa que el contenido de Wikipedia se puede distribuir y enlazar de acuerdo con lo establecido en estas licencias.

1. INTRODUCCIÓN

La Tienda de Robótica y el Equipo de Cosas de Mecatrónica traen esta guía que aborda el aprendizaje sobre el concepto DIY (Do it yourself) o en español "Hazlo tú mismo". Luego de una cuidadosa selección de componentes electrónicos y apoyados en la placa Arduino se crea el producto **Kit Básico de Arduino** apoyado de esta guía.

Abordamos temas fundamentales como el hardware y software libre, revisando de manera cuidadosa el proyecto Arduino y apoyándonos en el Software Fritzing para lograr montajes muy llamativos y semejantes a la realidad. No es necesario que sepas de electrónica y programación porque con los siguientes dos capítulos abordamos los conceptos desde lo más básico hasta lo fundamental. Luego de conocer estos conceptos tenemos un capítulo dedicado a que conozcas los componentes electrónicos como un LED, un motor, un buzzer y muchos más. Hemos diseñado un capítulo especial sobre Arduino y Fritzing los cuales no puedes dejar de ver. Este Kit Básico es ideal para todo ámbito de aprendizaje desde el colegio hasta universidades y si eres un entusiasta o un gomoso no puedes dejar de tener esta guía en casa, un capítulo completo se dedica a mostrar el kit de abajo a arriba :).

Finalmente llegamos a una parte muy especial, al capitulo de los tutoriales, donde paso a paso se explican 14 ejemplos, durante el recorrido de aprendizaje te encontrarás con preguntas, tips y ejercicios.

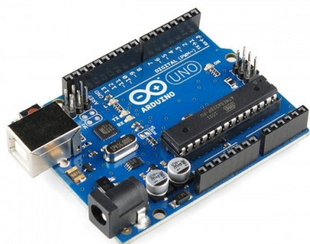
DIY

1A

Computación física

La **Computación física**, significa la construcción de sistemas interactivos físicos mediante el uso de software y hardware que pueden sentir y responder al mundo analógico. Si bien esta definición es suficientemente amplia para abarcar aspectos como los sistemas inteligentes de control de tráfico de automóviles o los procesos de automatización de fábricas, en un sentido más amplio, la computación física es un marco creativo para la comprensión de la relación de los seres humanos en el mundo digital. En la práctica, a menudo el término describe el arte hecho a mano, diseño de proyectos DIY o pasatiempos que utilizan sensores y microcontroladores para traducir entradas analógicas a sistemas basados en software, y/o controlar dispositivos electromecánicos como motores, servos, iluminación u otro hardware.

Otras implementaciones de computación física trabajan con el reconocimiento de la voz, la cual se capta e interpretan sus ondas sonoras a través de micrófonos u otros dispositivos de detección de ondas sonoras, también la visión por computador, que aplica algoritmos a los videos detectados por algún tipo de cámara. Interfaces táctiles son también un ejemplo de la computación física.



El prototipado (crear montajes rápidos con ayuda de una protoboard y componentes básicos de electrónica) juega un papel importante en la computación física. Herramientas como Arduino y Fritzing son útiles para diseñadores, artistas, estudiantes y entusiastas porque ayudan a elaborar prototipos rápidamente.



2. PARA TENER EN CUENTA

Te presentamos una información de interés que te recomendamos la tengas en cuenta para el desarrollo de la presente guía. Conoce más acerca de los desarrolladores y los pilares de este excelente material.

Tienda de Robótica

2A

Tienda de Robótica

La **Tienda de Robótica** nace en el 2010 con la idea de poner a disposición de estudiantes, profesionales y aficionados de la robótica los mejores productos usados en el mundo para desarrollar sus proyectos.

La responsabilidad social que el mundo nos exige hace que por medio de la Tienda de Robótica y el sitio web Cosas de Mecatrónica podamos compartir: noticias, eventos, tutoriales, proyectos entre otros. La Tienda de Robótica se encuentra ubicada en la ciudad de Bogotá D.C, Colombia y es solo virtual. Agradecemos que compres nuestros productos con el mejor precio del mercado. Contáctanos para conocer más de nosotros.

Cosas de Mecatrónica

2B



Cosas de Mecatrónica nació en Enero de 2007, después de muchos intentos de su creador Yesid Hernández de hacer páginas web. Hasta Abril de 2009 estuvo alojado en wordpress (http://mecatronica.wordpress.com). A partir de Mayo de 2009, Cosas de Mecatrónica adquirió un dominio propio con la idea de seguir compartiendo información y ayudar a crecer a la comunidad Mecatrónica en Español.

En principio nació solo como un sitio en donde se publicara y compartieran los proyectos mecatrónicos realizados por Ibraim Yesid Hernández Olarte, ahora busca ser un ambiente colaborativo de todas la personas y organizaciones interesadas en compartir información alrededor de la Mecatrónica. La participación en el BLOG viene principalmente de los alumnos del SENA, pero no sólo de ellos y se busca que se cree una comunidad compartiendo información de pequeños y grandes proyectos en los cuales todos estamos trabajando.

Hardware libre

2C



Se llama **hardware libre** a los dispositivos de hardware cuyas especificaciones y diagramas esquemáticos son de acceso público, ya sea bajo algún tipo de pago o de forma gratuita. La filosofía del software libre (las ideas sobre la libertad del conocimiento) es aplicable a la del hardware libre. Se debe recordar en todo momento que libre no es sinónimo de gratis. El hardware libre forma parte de la cultura libre.

Dado que el hardware tiene asociados a él costos variables directos, ninguna definición de software libre se puede aplicar directamente sin modificación. En cambio, el término hardware libre se ha usado principalmente para reflejar el uso del software libre con el hardware y el lanzamiento libre de la información con respecto al hardware, a menudo incluyendo el lanzamiento de los diagramas esquemáticos, diseños y montajes.



Software libre

2D



El **software libre** (en inglés free software, aunque esta denominación también se confunde a veces con "gratis" por la ambigüedad del término "free" en el idioma inglés, por lo que también se usa "libre software" y "logical libre") es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado, y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.



Creative commons

2E



Creative Commons (CC) es una organización no gubernamental sin ánimo de lucro que desarrolla planes para ayudar a reducir las barreras legales de la creatividad, por medio de nueva legislación y nuevas tecnologías. Las licencias Creative Commons o CC están inspiradas en la licencia GPL (General Public License) de la Free Software Foundation, compartiendo buena parte de su filosofía. La idea principal detrás de ellas es posibilitar un modelo legal ayudado por herramientas informáticas, para así facilitar la distribución y el uso de contenidos.

Existe una serie de licencias Creative Commons, cada una con diferentes configuraciones, que permite a los autores poder decidir la manera en la que su obra va a circular en internet, entregando libertad para citar, reproducir, crear obras derivadas y ofrecerla públicamente, bajo ciertas diferentes restricciones. **La licencia de la presente obra se expone en la página 16.**



Arduino



Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware libre, flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquier interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada, para esto toda una gama de sensores puede ser usada y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectarlo a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP).

Las placas pueden ser hechas a mano o comprarse montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, así pues eres libre de adaptarlos a tus necesidades.



www.arduino.cc

2F

Fritzing



Fritzing es un programa de automatización de diseño electrónico libre que busca ayudar a diseñadores y artistas para que puedan pasar de prototipos (usando, por ejemplo, placas de pruebas) a productos finales.

Fritzing fue creado bajo los principios de Processing y Arduino y permite a los diseñadores, artistas, investigadores y aficionados documentar sus prototipos basados en Arduino y crear esquemas de circuitos impresos para su posterior fabricación. Además, cuenta con un sitio web complementario que ayuda a compartir y discutir bosquejos, experiencias y a reducir los costos de fabricación.



www.fritzing.org

2G

Licencia de la obra

2H

Usted es libre de:

Compartir - copiar, distribuir, ejecutar y comunicar públicamente la obra

hacer obras derivadas

Bajo las condiciones siguientes:



Atribución — Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciante (pero no de una manera que sugiera que tiene su apoyo o que apoyan el uso que hace de su obra).



No Comercial — No puede utilizar esta obra para fines comerciales.



Compartir bajo la Misma Licencia — Si altera o transforma esta obra, o genera una obra derivada, sólo puede distribuir la obra generada bajo una licencia idéntica a ésta.

Entendiendo que:

Renuncia — Alguna de estas condiciones puede **no aplicarse** si se obtiene el permiso del titular de los derechos de autor

Dominio Público — Cuando la obra o alguno de sus elementos se halle en el **dominio público** según la ley vigente aplicable, esta situación no quedará afectada por la licencia.

Otros derechos — Los derechos siguientes no quedan afectados por la licencia de ninguna manera:

- Los derechos derivados de **usos legítimos**u otras limitaciones reconocidas por ley no se ven afectados por lo anterior.
- Los derechos **morales** del auto;
- Derechos que pueden ostentar otras personas sobre la propia obra o su uso, como por ejemplo **derechos de imagen** o de privacidad.

Aviso — Al reutilizar o distribuir la obra, tiene que dejar muy en claro los términos de la licencia de esta obra. La mejor forma de hacerlo es enlazar a esta página.

www.creativecommons.org/licenses/by-nc-sa/3.0/deed.es



3. ELECTRÓNICA

Desde el inicio de este capítulo te vas a enterar de los términos técnicos que tiene la electrónica y que al final de éste muy seguramente vas a manejar de manera experta.

Concepto de electrónica



Estudia y emplea sistemas cuyo funcionamiento se basa en la conducción y el control del flujo de los **electrones** u otras partículas cargadas eléctricamente.

El diseño y la gran construcción de circuitos electrónicos para resolver problemas prácticos forman parte de la electrónica y de los campos de la ingeniería electrónica, electromecánica y la informática en el diseño de software para su control.

3A

La electrónica desarrolla en la actualidad una gran variedad de tareas. Los principales usos de los circuitos electrónicos son el control, el procesado, la distribución de información, la conversión y la distribución de la energía eléctrica. Estos dos usos implican la creación o la detección de campos electromagnéticos y corrientes eléctricas.

Mira a tu alrededor radio, televisor, PC, teléfono móvil, lavadora todos ellos tienen electrónica.



Voltaje



Una magnitud física que impulsa a los electrones a lo largo de un conductor (por ejemplo un cable) en un circuito eléctrico cerrado, provocando el flujo de una corriente eléctrica. Su unidad es el Voltio(V). El instrumento usado para medir el voltaje se conoce como voltímetro.



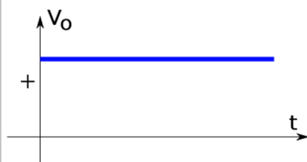
3B

Voltaje DC

3C

Es el flujo continuo de electrones a través de un conductor entre dos puntos de distinto potencial. En la corriente continua las cargas eléctricas circulan siempre en la misma dirección, es continua la corriente mantiene siempre la misma polaridad. En la norma sistemática europea el color negro corresponde al negativo y el rojo al positivo o sencillamente se simboliza para el positivo con VCC, +, VSS y para el negativo con 0V, -, GND.





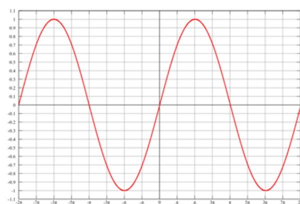
Muchos aparatos necesitan corriente continua para funcionar, sobre todos los que llevan electrónica (equipos audiovisuales, computadores, etc.), para ello se utilizan fuentes de alimentación. Lo puedes encontrar en la baterías, pilas, salida de los cargadores de computador.



Voltaje AC

3D

Es la corriente eléctrica en la que la magnitud y dirección varían cíclicamente. La forma de onda de la corriente alterna más comúnmente utilizada es la de una onda seno.



El voltaje AC es el que llega a la tomas de electricidad de los hogares y a las empresas, es muy común encontrarla en las tomas de corriente donde se conectan nuestros electrodomésticos. Sin embargo, las señales de audio y de radio transmitidas por los cables eléctricos son también ejemplos de corriente alterna. En estos usos, el fin más importante suele ser la transmisión y recuperación de la información codificada (o modulada) sobre la señal de la AC.





Corriente

3E

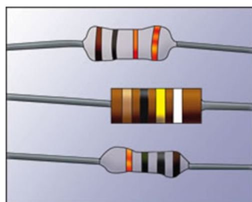
Es el flujo de electrones a través de un conductor o semiconductor en un sentido. La unidad de medida de ésta es el amperio (A). Una corriente eléctrica, puesto que se trata de un movimiento de cargas, produce un campo magnético, un fenómeno que puede aprovecharse en el electroimán, este es el principio de funcionamiento de un motor.



El instrumento usado para medir la intensidad de la corriente eléctrica es el galvanómetro que, calibrado en amperios, se llama amperímetro, colocado en serie con el conductor cuya intensidad se desea medir.



Resistencia

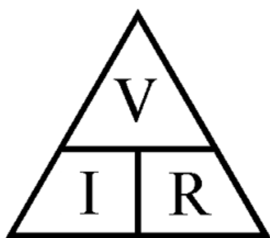


Es la propiedad física mediante la cual todos los materiales tienden a oponerse al flujo de la corriente. La unidad de este parámetro es el Ohmio (Ω). Puedes encontrar resistencias en los calefactores eléctricos, tarjetas electrónicas, estufas son muy útiles para limitar el paso de la corriente.



3F

Ley de Ohm



La ley dice que la corriente (I) que circula por un conductor eléctrico es directamente proporcional al voltaje (V) e inversamente proporcional a la resistencia (R).

La pirámide de la derecha es muy útil para conocer la fórmula a la que es igual la variable que tapes con el dedo, por ejemplo: Tapa con tu dedo la V (voltaje), entonces voltaje va a ser igual a I (corriente) por R (resistencia), una más, tapa I (Corriente), I va ser igual a V divido R.

3G

Sistemas electrónicos

Un sistema electrónico es un conjunto de circuitos que interactúan entre sí para obtener un resultado. Una forma de entender los sistemas electrónicos consiste en dividirlos en entradas, salidas y procesamiento de señal.



3H

Entradas

Las entradas o Inputs: Son sensores (o transductores) electrónicos o mecánicos que toman las señales (en forma de temperatura, presión, humedad, contacto, luz, movimiento, pH etc.) del mundo físico y las convierten en señales de corriente o voltaje.





Por ejemplo un sensor de temperatura, un pulsador, una fotocelda, un potenciómetro, un sensor de movimiento entre muchos más.



3I

3J

Salidas

Las salidas o Outputs: Son actuadores u otros dispositivos (también transductores) que convierten las señales de corriente o voltaje en señales físicamente útiles como movimiento, luz, sonido, fuerza, rotación entre otros.





Por ejemplo: un motor que gire, un LED o sistema de luces que se encienda automáticamente cuando esté oscureciendo, un buzzer que genere diversos tonos.



3K

Procesamiento de señal



Se realiza mediante circuitos de procesamiento de señales generalmente conocidos como microcontroladores. Consisten en piezas electrónicas conectadas juntas para manipular, interpretar y transformar las señales de voltaje y corriente provenientes de los sensores (Entradas) y tomar las respectiva decisiones para generar acciones en las salidas.



Resumen de los Sistemas Electrónicos

SISTEMAS ELECTRÓNICOS



Como ejemplo supongamos un televisor. Su entrada es una señal análoga recibida por una antena o por un cable. Los circuitos integrados del interior del televisor extraen la información sobre el brillo, el color y el sonido de esta señal. Los dispositivos de salida son una pantalla LED que convierte las señales electrónicas en imágenes visibles en una pantalla y unos altavoces.

Otro ejemplo puede ser el de un circuito que controle la temperatura de un lugar, el sensor de temperatura y el circuito integrado se encarga de convertir la señal de entrada en un nivel de voltaje apropiado y si la temperatura registrada es muy alta el circuito integrado envía la información a un motor para que este encienda el ventilador y refrigere el lugar.





Señales electrónicas

3L



Las entradas y salidas de un sistema electrónico serán consideradas como las señales variables. En electrónica se trabaja con variables que se toman en forma de voltaje o corriente, éstas se pueden denominar comúnmente señales.

Las señales primordialmente pueden ser de dos tipos descritos a continuación: Digital o análoga



Variable digital

3M



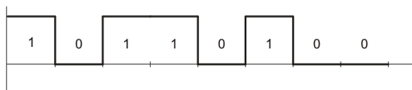
2 - 5v
Encendido
1
High



0 - 0.8v
Apagado
0
Low

También llamadas variables discretas. Se caracterizan por tener dos estados diferenciados y por lo tanto se pueden llamar binarias. Siendo estas variables más fáciles de tratar (en lógica serían los valores Verdadero (V) y Falso (F) o podrían ser 1 ó 0 respectivamente).

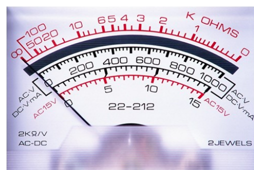
Un ejemplo de una señal digital es el interruptor del timbre de tu casa, por que este interruptor tiene dos estados pulsado y sin pulsar





Variable análoga

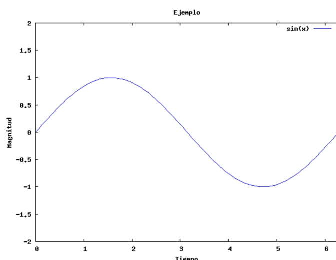
3N



Son aquellas que pueden tomar un número infinito de valores comprendidos entre dos límites. La mayoría de los fenómenos de la vida real son señales de este tipo. (sonido, temperatura, voz, video, etc.)

Un ejemplo de sistema electrónico analógico es un parlante, que se emplea para amplificar el sonido de forma que éste sea oído por una gran audiencia. Las ondas de sonido que son analógicas en su origen, son capturadas por un micrófono y convertidas en una pequeña variación analógica de tensión denominada señal de audio.





3O

Entrada / salida digital



Entrada
Pulsador



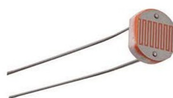
Salida
LED



Entrada
Reed switch

3P

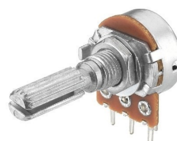
Entrada / salida análoga



Entrada
Fotocelda



Salida
Motor DC

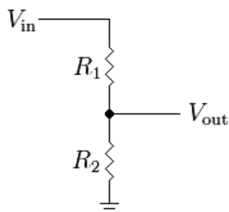


Entrada
Potenciómetro

3Q

Divisor de voltaje

Un divisor de voltaje es una configuración de circuito eléctrico que reparte el voltaje de una fuente (Vin) entre una o más resistencias (R1, R2) conectadas en serie (una a continuación de otra).

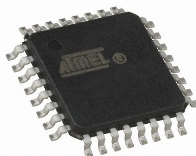




$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in}$$

3R

Conversor análogo-digital CAD ó ADC



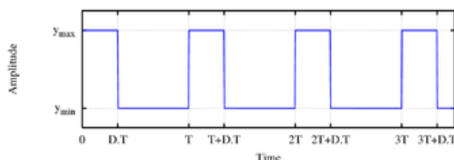
Un conversor (o convertidor) analógico-digital (CAD), (o también ADC del inglés "Analog-to-Digital Converter") es un dispositivo electrónico capaz de convertir una entrada analógica de voltaje en un valor binario. Se utiliza en equipos electrónicos como computadores, grabadores de sonido y de vídeo, y equipos de telecomunicaciones. La señal analógica, que varía de forma continua en el tiempo, se conecta a la entrada del dispositivo y se somete a un muestreo a una velocidad fija, obteniéndose así una señal digital a la salida del mismo.



Módulación por ancho del pulso PWM

La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica (una senoidal o una cuadrada, por ejemplo), ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

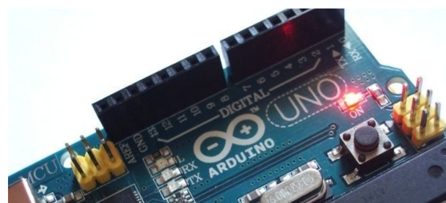
Por ejemplo si le aplicamos PWM a un LED podemos variar su intensidad de brillo y si le aplicamos un PWM a un motor DC logramos variar la velocidad del mismo con la característica de mantener su par (fuerza) constante.





3S

Comunicación serial



Es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadores y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez. Uno de sus usos es monitorear a través de la pantalla del computador el estado del periférico conectado, por ejemplo al pulsar la letra A en el teclado se debe encender un LED conectado de manera remota al computador.



3T

Hoja dejada en blanco de manera intencional

4. COMPONENTES ELECTRÓNICOS

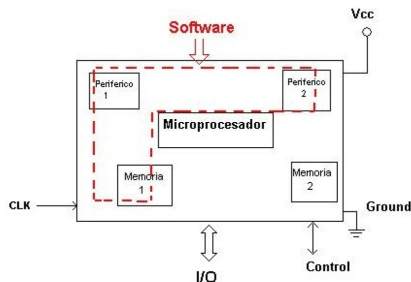
Diversos componentes electrónicos unen sus fuerzas para lograr aplicaciones fantásticas como por ejemplo el televisor de tu casa o el computador, por dentro de ellos vas e encontrar tarjetas con resistenticas, condensadores, circuitos integrados, transistores entre otros.

4A

Microcontrolador



Un microcontrolador (abreviado µC,UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres unidades funcionales principales: unidad central de procesamiento (CPU), memoria y periféricos de entrada y salida (I/O).



Para que pueda controlar algún proceso es necesario crear y luego grabar en la memoria EEPROM del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores; debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

Los microcontroladores representan la inmensa mayoría de los chips vendidos, tienes distribuidos seguramente entre los electrodomésticos de tu hogar una o dos docenas de microcontroladores. Pueden encontrarse en casi cualquier dispositivo electrónico como automóviles, lavadoras, hornos microondas, teléfonos, Arduino.

Los microcontroladores utilizan la mayoría para recibir señales de dispositivos de entrada/salida, con la gran ventaja de que se puede prescindir de cualquier otra circuitería externa.

PDIP

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

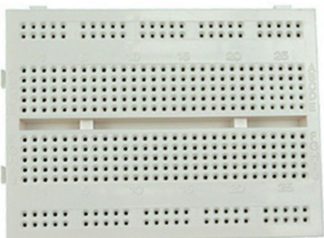
Los puertos de E/S (entrada/salida ó I/O) en el microcontrolador, se agrupan en puertos de 8 bits de longitud, lo que permite leer datos del exterior o escribir en ellos desde el interior del microcontrolador, el destino es el trabajo con dispositivos simples como relés, LED, motores, fotoceldas, pulsadores o cualquier otra cosa que se le ocurra al programador.



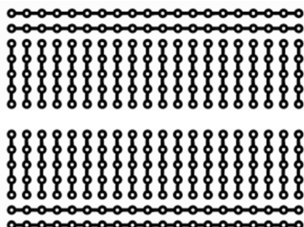
4B

Protoboard

Es una placa reutilizable usada para construir prototipos de circuitos electrónicos sin soldadura. Compuestas por bloques de plástico perforados y numerosas láminas delgadas de una aleación de cobre, estaño y fósforo



Vista real



Conexiones internas

4C

Resistencia



Símbolo



Componente

Es un material formado por carbón y otros elementos resistivos para disminuir la corriente que pasa. Se opone al paso de la corriente. La corriente máxima en un resistor viene condicionado por la máxima potencia que puede disipar su cuerpo. Esta potencia se puede identificar visualmente a partir del diámetro sin que sea necesaria otra indicación. Los valores más comunes son 0,25 W, 0,5 W y 1 W.

El valor de la resistencia eléctrica se obtiene leyendo las cifras como un número de una, dos o tres cifras; se multiplica por el multiplicador y se obtiene el resultado en Ohmios (Ω).



QUIZ

Completa de acuerdo a la tabla de colores de la siguiente página

1- ______Ω



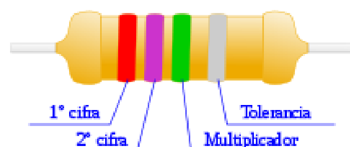
2- ______Ω



Tabla de colores de las resistencias

Color de la banda		Valor de la 1°cifra significativa	Valor de la 2°cifra significativa	Multiplicador	Tolerancia
Negro		-	0	1	-
Marrón		1	1	10	±1%
Rojo		2	2	100	±2%
Naranja		3	3	1 000	-
Amarillo		4	4	10 000	±4%
Verde		5	5	100 000	±0,5%
Azul		6	6	1 000 000	±0,25%
Violeta		7	7	10000000	±0,1%
Gris		8	8	100000000	±0.05%
Blanco		9	9	1000000000	-
Dorado		-	-	0,1	±5%
Plateado		-	-	0,01	±10%
Ninguno		-	-	-	±20%

Ejemplo



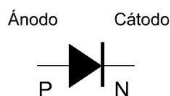
La caracterización de una resistencia de 2.700.000 Ω (2,7 MΩ), con una tolerancia de ±10%, sería la representada en la figura :

- 1ª cifra: rojo (2)
- 2ª cifra: violeta (7)
- Multiplicador: verde (100000)
- Tolerancia: plateado (±10%)



4E

Diodo





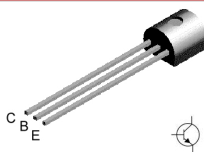
Símbolo Componente

Un diodo es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él en un solo sentido. Tiene dos partes: el cátodo y el ánodo.

4F

Transistor





Símbolo Componente

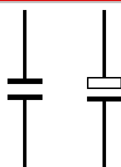
El transistor es un dispositivo electrónico semiconductor que cumple funciones de amplificador, oscilador, conmutador o rectificador. Tiene tres partes: la base (B), el emisor (E) y colector (C).

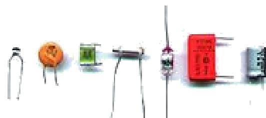
Actualmente se encuentran prácticamente en todos los aparatos domésticos de uso diario: radios, televisores, grabadoras, reproductores de audio y video, hornos de microondas, lavadoras, automóviles, equipos de refrigeración, alarmas, relojes de cuarzo, ordenadores, calculadoras, impresoras, lámparas fluorescentes, equipos de rayos X, tomógrafos, ecógrafos, reproductores mp3, teléfonos celulares, etc.



4G

Condensador



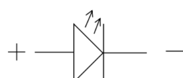


Símbolo Componente

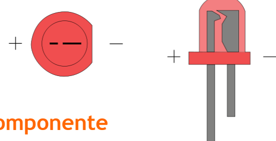
Un condensador o capacitor es un dispositivo utilizado en electrónica, capaz de almacenar energía sustentando un campo eléctrico. Está formado por un par de superficies conductoras, generalmente en forma de láminas o placas, en situación de influencia total separadas por un material dieléctrico o por el vacío. Las placas, sometidas a una diferencia de potencial, adquieren una determinada carga eléctrica, positiva en una de ellas y negativa en la otra.



LED



Símbolo



Componente

Un LED (Diodo emisor de luz, también "diodo luminoso") es un diodo semiconductor que emite luz. Se usan como indicadores en muchos dispositivos, y cada vez con mucha más frecuencia en iluminación. Los LEDs presentan muchas ventajas sobre las fuentes de luz incandescente como un consumo de energía mucho menor, mayor tiempo de vida, menor tamaño, gran durabilidad y fiabilidad.

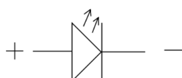


El LED tiene una polaridad, un orden de conexión, y al conectarlo al revés se puede quemar, revisa los dibujos de la parte superior para conocer a que corresponde el positivo y el negativo.

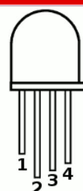


4H

LED RGB



Símbolo



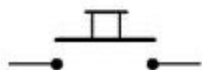
Componente



Un LED RGB es un LED que incorpora en su mismo encapsulado tres LEDs, es RGB porque R (red, rojo), G (green, verde) y B (blue, azul) así se pueden formar miles de colores ajustando de manera individual cada color. Los tres LEDs están unidos por el negativo o cátodo.

4I

Pulsador



Símbolo



Componente

Un botón o pulsador es utilizado para activar alguna función. Los botones son por lo general activados al ser pulsados, normalmente con un dedo. Un botón de un dispositivo electrónico funciona por lo general como un interruptor eléctrico, es decir en su interior tiene dos contactos, si es un dispositivo NA (normalmente abierto) o NC (normalmente cerrado), con lo que al pulsarlo se activará la función inversa de la que en ese momento este realizando.



4J

4K

Reed switch





Símbolo

Componente

Reed switch es un interruptor eléctrico activado por un campo magnético, por ejemplo con un imán. Cuando los contactos están normalmente abiertos se cierran en la presencia de un campo magnético; cuando están normalmente cerrados se abren en presencia de un campo magnético. Un uso muy extendido se puede encontrar en los sensores de las puertas y ventanas de las alarmas antirobo, el imán va unido a la puerta y el reed switch al marco.



4L

Potenciómetro





Símbolo

Componente

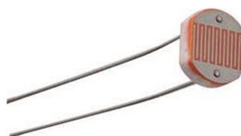
Un potenciómetro es una resistencia cuyo valor de resistencia es variable. De esta manera, indirectamente, se puede controlar la intensidad de corriente que fluye por un circuito si se conecta en paralelo, o controlar el voltaje al conectarlo en serie. Son adecuados para su uso como elemento de control en los aparatos electrónicos. El usuario acciona sobre ellos para variar los parámetros normales de funcionamiento. Por ejemplo, el volumen de un radio.



4M

Fotocelda





Símbolo

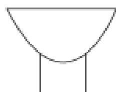
Componente

Es una resistencia, cuyo valor en ohmios varía ante las variaciones de la luz incidente. Una fotocelda presenta un bajo valor de su resistencia ante la presencia de luz y un alto valor de resistencia ante la ausencia de luz.

Pueden encontrarse en muchos artículos de consumo, como por ejemplo en cámaras, medidores de luz, relojes con radio, alarmas de seguridad o sistemas de encendido y apagado del alumbrado público de las calles.



Zumbador o buzzer



Símbolo



Componente

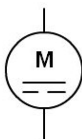
El zumbador, buzzer en inglés, es un transductor electroacústico que produce un sonido o zumbido continuo o intermitente de un mismo tono. Sirve como mecanismo de señalización o aviso, y son utilizados en múltiples sistemas como en automóviles o en electrodomésticos. Inicialmente este dispositivo estaba basado en un sistema electromecánico que era similar a una campana eléctrica pero sin el badajo metálico, el cual imitaba el sonido de una campana.

Su construcción consta de dos elementos, un electroimán y una lámina metálica de acero. El zumbador puede ser conectado a circuitos integrados especiales para así lograr distintos tonos. Cuando se acciona, la corriente pasa por la bobina del electroimán y produce un campo magnético variable que hace vibrar la lámina de acero sobre la armadura.



4N

Motor DC



Símbolo



Componente

El motor de corriente continua (DC) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio. Esta máquina de corriente continua es una de las más versátiles. Su fácil control de posición, paro y velocidad la han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos. Por ejemplo, los puedes encontrar en la tracción de los carros de juguetes de pilas o en las llantas de los robots.



4O

Hoja dejada en blanco de manera intencional

5. PROGRAMACIÓN

La programación es un gran recurso que nos permite crear diversas secuencias de pasos lógicos que van a satisfacer nuestras necesidades y las de nuestros sistemas. Programar es todo un arte que requiere de una gran habilidad lógica y concentración por parte del programador.

Concepto de programación



Es el proceso de diseñar, escribir, probar, depurar y mantener el código fuente de programas computacionales. El código fuente es escrito en un lenguaje de programación. El propósito de la programación es crear programas que exhiban un comportamiento deseado.

5A

El proceso de escribir código requiere frecuentemente conocimientos en varias áreas distintas, además del dominio del lenguaje a utilizar, algoritmos especializados y lógica formal. Programar involucra áreas como el análisis y diseño de la aplicación.

Para crear un programa que el computador interprete y ejecute las instrucciones escritas en él, debe usarse un **Lenguaje de programación**. En sus inicios los computadores interpretaban sólo instrucciones en un lenguaje específico, del más bajo nivel conocido como código máquina, siendo éste excesivamente complicado para programar. De hecho sólo consiste en cadenas de números 1 y 0 (Sistema binario).

Para facilitar el trabajo de programación, los primeros científicos que trabajaban en el área decidieron reemplazar las instrucciones, secuencias de unos y ceros, por palabras o letras provenientes del inglés, codificándolas así y creando un lenguaje de mayor nivel, que se conoce como Assembly o lenguaje ensamblador. Por ejemplo, para sumar se usa la letra A de la palabra inglesa add (sumar). En realidad escribir en lenguaje ensamblador es básicamente lo mismo que hacerlo en lenguaje máquina, pero las letras y palabras son bastante más fáciles de recordar y entender que secuencias de números binarios.

A medida que la complejidad de las tareas que realizaban las computadoras aumentaba, se hizo necesario disponer de un método sencillo para programar. Entonces, se crearon los lenguajes de alto nivel. Mientras que una tarea tan trivial como multiplicar dos números puede necesitar un conjunto de instrucciones en lenguaje ensamblador, en un lenguaje de alto nivel bastará con sólo una.



Lenguaje de programación

5B

```
def add5(x):
   return x+5

def dotwrite(ast):
   nodename = getNodename()
   label=symbol.sym_name.get(int(ast[0]),ast[0])
   print '    %s [label="%s' % (nodename, label),
   if isinstance(ast[1], str):
      if ast[1].strip():
         print '= %s"];' % ast[1]
      else:
         print '"]'
    else:
      print '"];'
      children = []
      for  in n, childenumerate(ast[1:]):
         children.append(dotwrite(child))
      print ,'    %s -> {' % nodename
      for  in :namechildren
         print '%s' % name,
```

Un lenguaje de programación es un idioma artificial diseñado para expresar operaciones que pueden ser llevadas a cabo por máquinas como los computadores. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.



Lenguaje máquina

5C



El Lenguaje de máquina es el sistema de códigos directamente interpretable por un circuito microprogramable, como el microprocesador de un computador o un microcontrolador. Este lenguaje está compuesto por un conjunto de instrucciones que determinan acciones a ser tomadas por la máquina. Estas instrucciones son normalmente ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos.

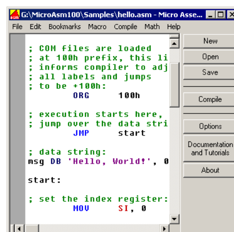
El lenguaje máquina trabaja con dos niveles de voltaje. Dichos niveles, por abstracción, se simbolizan con el cero (0) y el uno (1), por eso el lenguaje de máquina sólo utiliza dichos signos. Esto permite el empleo de las teorías del álgebra booleana y del sistema binario en el diseño de este tipo de circuitos y en su programación.



Lenguaje emsamblador

5D

El lenguaje ensamblador o assembler es un lenguaje de programación de bajo nivel para los computadores, microcontroladores, y otros circuitos integrados programables. Implementa una representación simbólica de los códigos de máquina. Esta representación es usualmente definida por el fabricante de hardware, y está basada en códigos mnemotécnicos que simbolizan los pasos de procesamiento (las instrucciones).





Un lenguaje ensamblador es por lo tanto específico a cierta arquitectura de computador física (o virtual). Esto está en contraste con la mayoría de los lenguajes de programación de alto nivel que idealmente son portables.

Lenguaje de alto nivel



Un lenguaje de programación de alto nivel se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad ejecutora de las máquinas.

Para los lenguajes de alto nivel se requiere de ciertos conocimientos de programación para realizar las secuencias de instrucciones lógicas. Los lenguajes de alto nivel se crearon para que el usuario común pudiese solucionar un problema de procesamiento de datos de una manera más fácil y rápida.

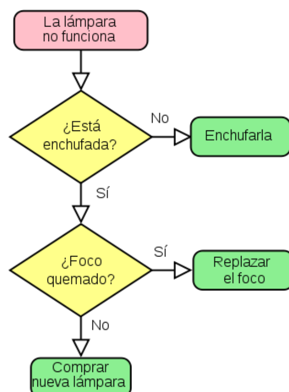


5E

Algoritmo

Un algoritmo es un conjunto pre-escrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad. Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.



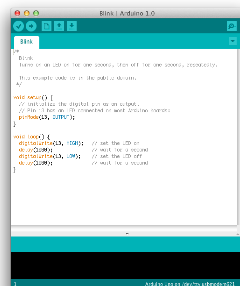


5F

Cuerpo de un programa en Arduino

Arduino se programa en el lenguaje de alto nivel C/C++ y generalmente tiene los siguiente componentes para elaborar el algoritmo:

- Estructuras
- Variables
- Operadores matemáticos, lógicos y booleanos
- Estructuras de control (Condicionales y ciclos)
- Funciones

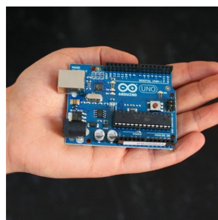


5G

Estructuras

5H

Son dos funciones principales que debe tener todo programa en Arduino:




```
setup(){
}
```

Código de configuración inicial, solo se ejecuta una vez.

```
loop(){
}
```

Esta función se ejecuta luego del setup(), se mantiene ejecutándose hasta que se des- energice o desconecte el Arduino.

Variables

5I

Es un dato o conjunto de datos que cambia su valor con la ejecución del programa.

Booleano	Entero	Carácter
true ó false	Valor entero	Almacena un ASCII
Boolean encendido=true;	int conta=5;	char letra='a';

Estos son algunos tipos de variables y los más utilizados en esta guía. Para más tipos visita:

arduino.cc/en/Reference/HomePage

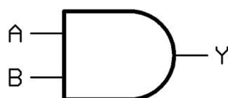
Operadores booleanos

5J

Usados generalmente dentro del condicional **If**

- && (y)
- || (o)
- ! (negación)

If (a || b)



Operadores de comparación

5K

Usados generalmente dentro del condicional **If y sobre el For y While**

- == (igual a)
- != (diferente de)
- < (menor que)
- > (mayor que)
- <= (menor o igual)
- >= (mayor o igual)



If (a == b)

5L

Operadores matemáticos

Se aplican al manejo de variables, condicionales y ciclos

- = (asignar)
- % (módulo)
- + (suma)
- - (resta)
- * (multiplicación)
- / (división)



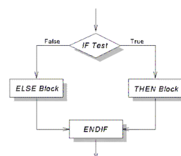
```
int valor = valor +5
```

5M

Estructuras de control

Son instrucciones que nos permiten tomar decisiones y hacer diversas repeticiones de acuerdo a unos parámetros, dentro de las más importantes podemos destacar:

- If
- Switch/case
- For
- While



5N

Condicionales

Ideales para tomar decisiones luego de evaluar las condiciones lógicas:

If (Si)

```
if (entrada < 500)
{
        // acción A
} else
{
        // acción B
}
```

Switch/case (Casos)

```
switch (var) {
  case 1:
    // acción A
    break;
  case 2:
    // acción B
    break;
  default:
    // acción C
}
```

5O

Ciclos

Ideales para repetir lo que se encuentre dentro de ellos

For (por)

```
for( int a=0; a>10; a++ )
{
        // acción a repetir
}
```

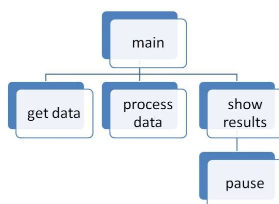
While (mientras)

```
while ( var < 200) {
  // acción a repetir
  var++;
}
```

5P

Funciones

Una función es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento. Las funciones son utilizadas para descomponer grandes problemas en tareas simples y para implementar operaciones que son comúnmente utilizadas durante un programa y de esta manera reducir la cantidad de código.



Cuando una función es invocada se le pasa el control a la misma, una vez que ésta finalizó con su tarea el control es devuelto al punto desde el cual la función fue llamada.

5Q

Funciones digitales

Orientas a revisar el estado y la configuración de las entradas y salidas digitales

pinMode()

Permite configurar un pin

pinMode(pin,modo)

```
pinMode (13,OUTPUT);
pinMode (a,INPUT);
```

digitalRead()

Leer un pin digital (0 ó 1)

digitalRead(pin)

```
int a = digitalRead (13);
```

digitalWrite()

Escribir un pin digital con 1 ó 0

digitalWrite(pin,estado)

```
digitalWrite (13,HIGH);
digitalWrite (13,LOW);
```

QUIZ

Completa de acuerdo a los conceptos acerca de programación

1- El lenguaje máquina se escribe en:

____

2- El If es un tipo de:

____



Funciones análogas

Ideales para la lectura y escritura de valores análogos

analogRead()

Leer un valor análogo 0 a 1023

analogRead(pin)

```
int a = analogRead (A0);
```

analogWrite() —> PWM

Escribir un valor análogo 0 a 255

analogWrite(pin,valor de PWM)

```
analogWrite (9, 134);
```

EJERCICIOS

1 **Relaciona los siguientes términos, cada letra tiene tres términos asociados:**

A- Carbón	Amperio	Flujo de electrones
B- Emisor de luz	LED	Verde
C- Lenguaje ensamblador	analogWrite()	Software libre
D- Digital	Hardware libre	PWM
E- Corriente	Resistencia	Pulsador
F- Arduino	ADD	Instrucción
G- Análogo	digitalRead()	1KΩ



5S Referencia rápida para programar

El programa de Arduino se puede dividir en tres partes principales: la estructura, las variables (valores y constantes) y funciones.

ESTRUCTURA

- setup()
- loop()

+Estructuras de control
- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

+Sintaxis
- ; (punto y coma)
- { } (llaves)
- / / (comentario de una sola línea)
- / * * / (comentario de varias líneas)
- # define
- # include

+Operadores matemáticos
- = (operador de asignación)
- + (suma)
- - (resta)
- * (multiplicación)
- / (división)
- % (módulo)

+Operadores de comparación
- == (igual que)
- ! = (diferente de)
- < (menor que)
- > (mayor que)
- <= (menor o igual a)
- >= (mayor o igual a)

+Operadores booleanos
- && (y)
- | | (o)
- ! (no)

+Acceso con apuntadores
- * eliminar la referencia del operador
- & operador de referencia

+Operadores bit a bit
- & (bit a bit AND)
- | (bit a bit OR)
- ^ (bit a bit XOR)
- ~ (bit a bit NOT)
- << (a la izquierda BitShift)
- >> (a la derecha BitShift)

+Operadores compuestos
- + + (incremento)
- -- (decremento)
- + = (compuesto adición)
- - = (compuesto substracción)
- * = (compuesto multiplicación)
- / = (compuesto división)
- & = (compuesto bit a bit AND)
- | = (compuesto bit a bit OR)

VARIABLES

+Constantes
- HIGH | LOW
- INPUT | OUTPUT
- true | false
- Constantes enteras
- Constantes flotante

+Tipos de datos
- void
- boolean
- char
- byte
- int
- word
- long
- unsigned long
- float
- double
- string - arreglo char
- String - objeto
- array

+Utilidades
- sizeof()

+Conversión
- char()
- byte()
- int()
- word()
- long()
- float()

FUNCIONES

+Digital I/O
- pinMode()
- digitalWrite()
- digitalRead()

+Analogo I/O
- analogReference()
- analogRead()
- analogWrite() - PWM

+Avanzadas I/O
- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

+Tiempo
- millis()
- micros()
- delay()
- delayMicroseconds()

+Matemáticas
- min()
- max()
- abs()
- constrain()
- map()
- pow()
- sqrt()

+Trigonométricas
- sin()
- cos()
- tan()

+Números aleatorios
- randomSeed()
- random()

+Bits y Bytes
- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()

+Interrupciones externas
- attachInterrupt()
- detachInterrupt()

+Interrupciones
- interrupts()
- noInterrupts()

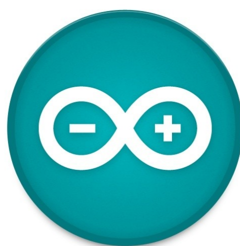
+Comunicación
- Serial
    - begin()
    - end()
    - available()
    - read()
    - peek()
    - flush()
    - print()
    - println()
    - write()

LIBRERÍAS

- EEPROM - leer y escribir
- Ethernet - conectarse a Internet
- Cristal líquido - control de LCD
- SD - lectura y escritura de tarjetas SD
- Servo - control de servomotores
- SPI - comunicación por el bus SPI
- Paso a paso - control de motores
- Wire - enviar y recibir datos TWI/I2C

... y muchas más visita

arduino.cc



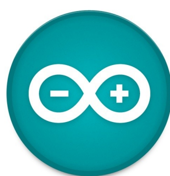
Hoja dejada en blanco de manera intencional

6. ARDUINO

Con las capacidades de Arduino solo debemos tener nociones básicas de electrónica y programación, eso es suficiente para comenzar a desarrollar nuestros proyectos. Arduino cuenta con una gran comunidad donde se comparte todo lo desarrollado y es una gran ventana para que puedas ver todo lo que es posible desarrollar.

6A

Proyecto Arduino



El proyecto comenzó en Ivrea, Italia (el sitio de la compañía de computadoras Olivetti), en el año 2005 con el fin de crear un dispositivo para estudiantes para el control integrado de proyectos de diseño e interacción, con la finalidad de que fuera más barato que los sistemas de creación de prototipos disponibles en ese entonces. A partir de mayo de 2011, más de 300.000 unidades de Arduino han sido distribuidas. Los fundadores Massimo Banzi y David Cuartielles nombraron el proyecto como Arduino de Ivrea, un protagonista histórico de la ciudad. En primer lugar "Arduino" es un termino masculino italiano, que significa "gran amigo".

El proyecto Arduino es un fork (en la ingeniería de software, un fork es un proyecto que sucede cuando los desarrolladores tienen una copia legal del código fuente y empiezan el desarrollo independiente de ella, creando una obra distinta de software) de la plataforma Wiring de código abierto. Wiring fue creado por el artista colombiano y programador Hernando Barragán como una tesis de maestría en el Instituto de diseño e interacción Ivrea, bajo la supervisión de Massimo Banzi y Casey Reas. Por otra parte, Wiring se basa en Processing y su entorno de desarrollo integrado creado por Casey Reas y Ben Fry.





"Arduino fue construido en torno al proyecto Wiring de Hernando Barragán. Wiring fue el proyecto de tesis de Hernando en el Instituto de diseño e interacción Ivrea. Fue pensado para ser una versión electrónica de Processing que utiliza nuestro entorno de programación y fue modelado para la sintaxis de Processing. Fue supervisado por mí mismo y Massimo Banzi, un fundador de Arduino. No creo que Arduino existiría sin Wiring y no creo que Wiring existiría sin Processing. Y sé que Processing sin duda no existiría sin Design By Numbers y John Maeda[1]"

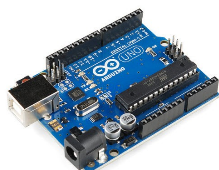
1 Entrevista con Ben Fry y Casey Reas por Shiffman Daniel (Sep 23, 2009), ver más



Familia Arduino

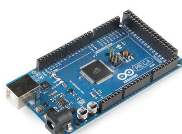
Como toda familia que se respete, tenemos de los integrantes más grandes a los más pequeños, te invitamos a conocerlos:

Arduino UNO



El más vendido, todo en uno

Arduino Mega2560



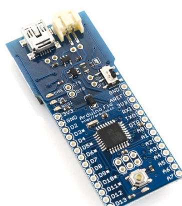
Muchas entradas y salidas

Arduino Pro



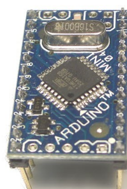
Ligero y práctico

Arduino Fio



Programación XBee

Arduino Mini



Pequeño, ideal para protoboard

Arduino Bluetooth



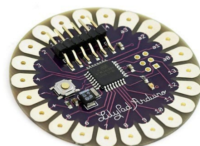
Programación vía bluetooth

Arduino Mega ADK



Para conectar el Android

Arduino LyliPad



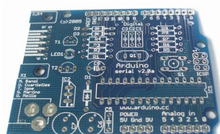
Para artistas y manejar e-textils

Arduino Nano



Un pequeño poderoso

Arduino Serial



El inicio del proyecto Arduino

Arduino Leonardo



El más joven y económico

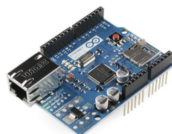
Expandir Arduino con los shields

Un **Shield** o escudo es una placa que permite expandir funcionalidades a tu Arduino, con lo cual puedes conectar motores, o a la red celular, a una red WiFi, a una red Ethernet o tener un MP3 en el Arduino, entre muchos más solo mira:

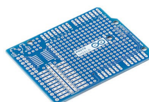
Celular



Ethernet



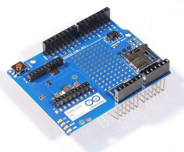
Proto



GPS



XBee



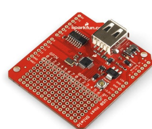
WiFi



LCD a color



USB host



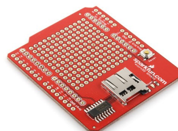
Joystick



Motores DC

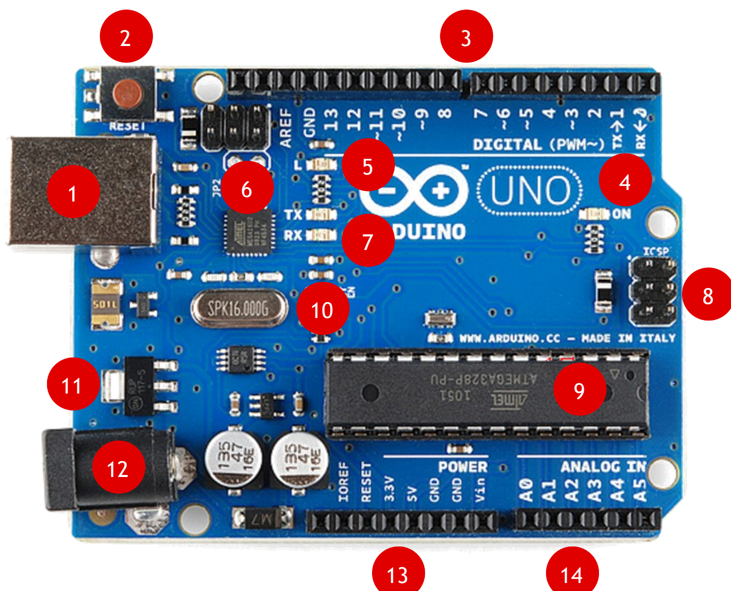


SD Card

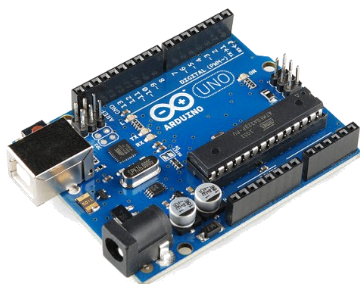


6D Placa Arduino Uno y sus partes

Vista frontal



Vistas auxiliares





DESCRIPCIÓN DE PARTES

1. Conector USB para el cable Tipo AB
2. Pulsador de Reset
3. Pines de E/S digitales y PWM
4. LED verde de placa encendida
5. LED naranja conectado al pin13
6. ATmega 16U2 encargado de la comunicación con el PC
7. LED TX (Transmisor) y RX (Receptor) de la comunicación serial
8. Puerto ICSP para programación serial
9. Microcontrolador ATmega 328, cerebro del Arduino
10. Cristal de cuarzo de 16Mhz
11. Regulador de voltaje
12. Conector hembra 2.1mm con centro positivo
13. Pines de voltaje y tierra
14. Entradas análogas

6E Instalando drivers

MAC y LINUX

Si tu computador tiene de sistema operativo alguna versión de Mac o una distribución de LINUX, lo único que debes hacer es:

1. Conectar la placa Arduino Uno al PC
2. Descargar el software de **arduino.cc/en/Main/Software**
3. Listo para trabajar y cargar programas

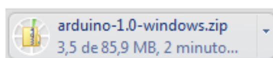



WINDOWS 7, Vista y XP

Si tu computador tiene de sistema operativo Windows en versión 7, Vista o XP, debes realizar la siguiente sucesión de sencillos pasos:

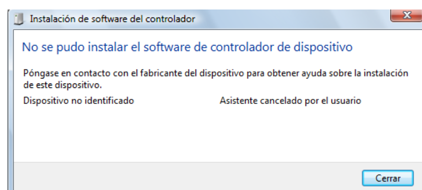
Windows

1. Descargar el software de **arduino.cc/en/Main/Software** para Windows

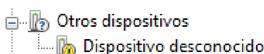


2. Descomprimir la carpeta de Arduino en una ubicación de fácil acceso

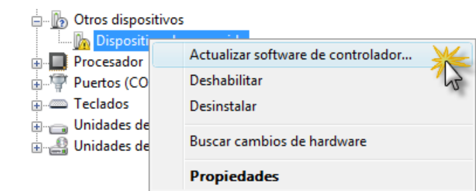
3. Conectar la placa Arduino Uno al PC y ver este aviso. No nos debemos preocupar



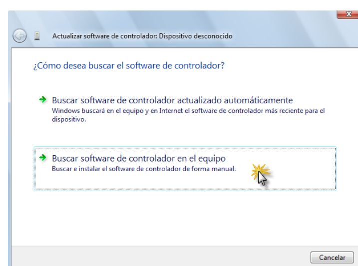
4. Visitar **Panel de control** y luego **Administrador de dispositivos**, allí buscar la siguiente opción



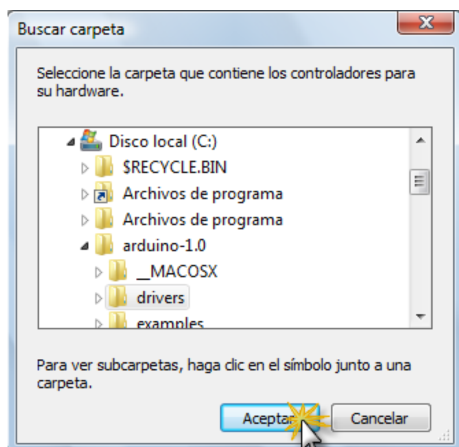
5 Click derecho sobre **Dispositivo desconocido** y luego sobre la opción **Actualizar software del controlador.**



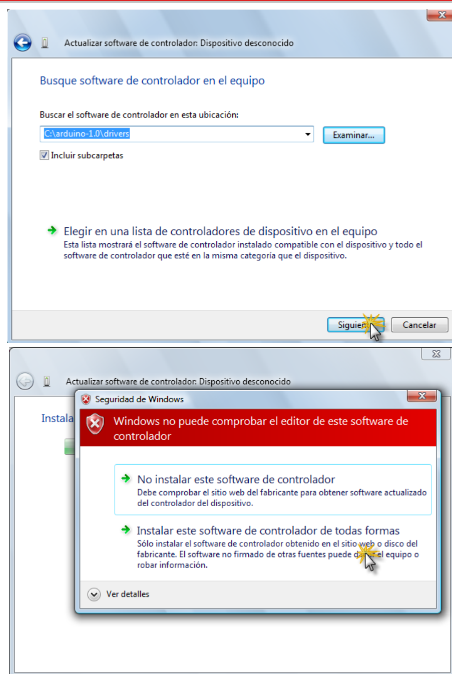
6 Ingresar a la opción Buscar **software de controlador en el equipo**



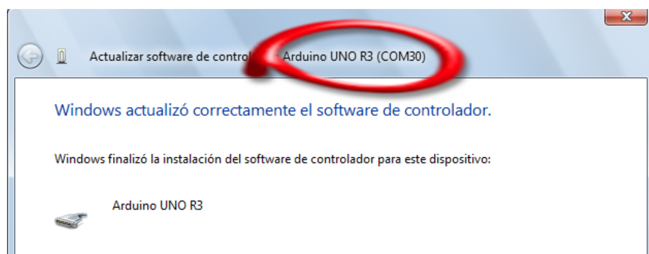
7 Examinar y buscar la **carpeta de Arduino** previamente descomprimida en el paso 2. Dentro de esa carpeta acceder a la carpeta Drivers y dar **Aceptar**



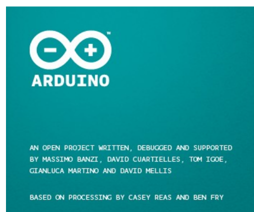
8 Una vez buscamos la carpeta de **Drivers** le damos **Siguiente**



9 Recibimos la confirmación del **Puerto COM** asignado, este número de Puerto COM es muy importante tenerlo en cuenta a la hora de programar.



Conociendo el software Arduino





Para ejecutar el programa Arduino, ingresamos a la carpeta de Arduino y allí buscamos el icono ce Arduino y le damos doble click





Compilar
Cargar a la placa
Nuevo
Abrir
Guardar

sketch_jan16a | Arduino 1.0

File Edit Sketch Tools Help

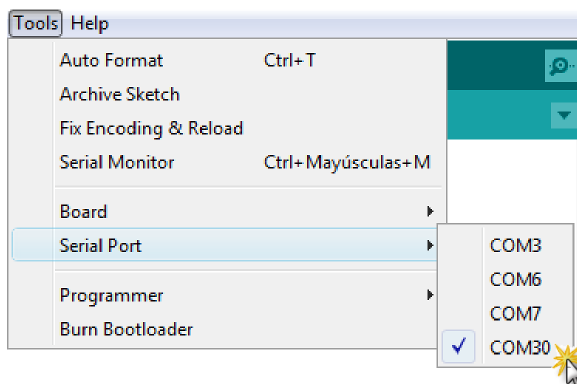
sketch_jan16a

Zona para escribir el código

Zona de mensajes del software
Errores y acciones

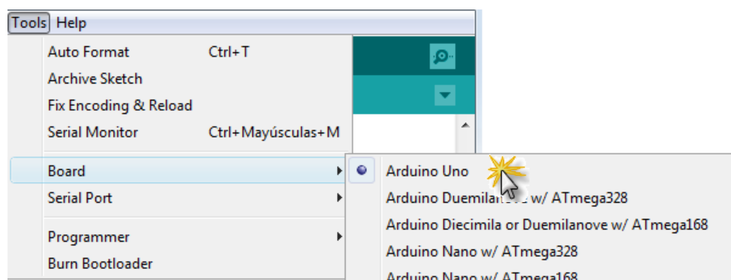
1 Arduino Uno on COM30

1 Puerto COM

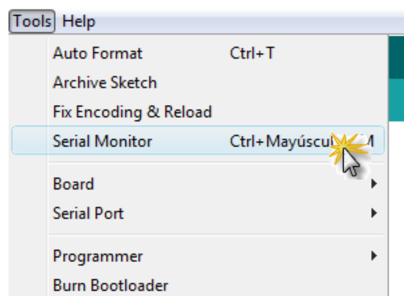


6F

2 Seleccionar la placa a trabajar



3 Consola serial

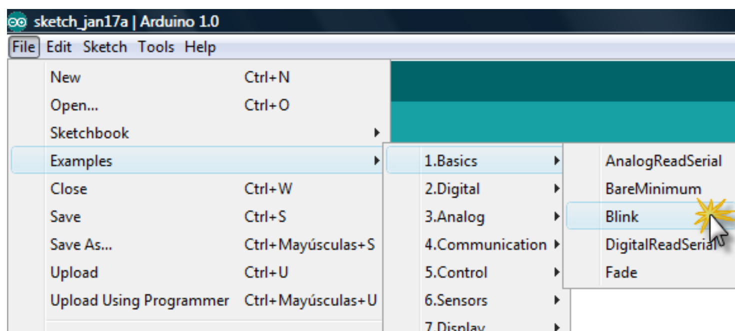


Cargando mi primer programa

Al final de este ejercicio ya vas a tener la capacidad de cargar programas a tu placa Arduino. Para ello abre el software de Arduino y realiza lo que se indica en la siguiente imagen.

6G

1



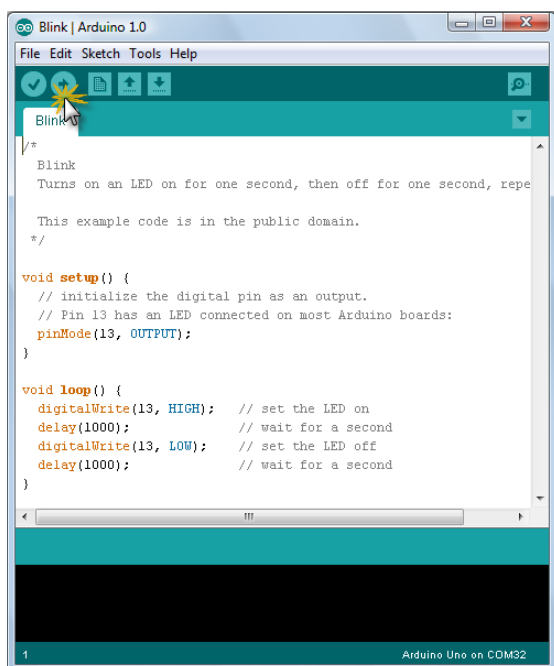
Cargando mi primer programa

Antes de continuar con el siguiente paso asegúrate de configura de manera correcta:

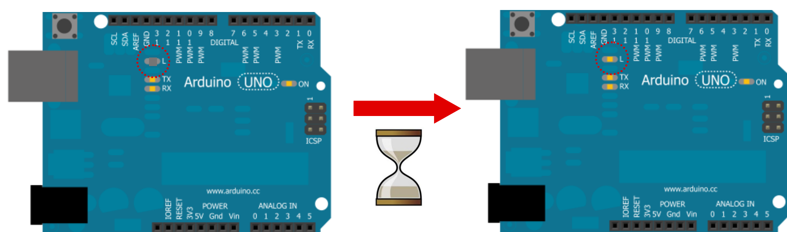
A- **Puerto COM**, revisa el **Paso 1** de la sección **6F**, recuerda que el valor del puerto COM lo obtuvimos en el **Paso 9** de la sección **6E**

B- **Board,** revisa el **Paso 2** de la sección **6F**, recuerda que para este caso la placa es **Arduino UNO**

2



3 El programa de intermitencia (Blink) que acabas de cargar en electrónica se llama **"Hola mundo",** consiste en prender y apagar un LED en intervalos de un segundo. El LED que prende y apaga es la parte **5** según la sección **6D** o el marcado con la letra **L** según la imagen de abajo. Ahora te podemos dar la **¡Bienvenid@ al mundo de Arduino :D!**



Ejercicio

A partir del ejemplo cargado en la sección anterior, ahora te proponemos que modifiques un poco el programa, para ello en las dos líneas de código donde dice:

```
delay(1000);
```

Cambia el valor de 1000 por 2000 y vuelve a cargar el programa a tu placa Arduino Uno, ¿que observas?



7. KIT BÁSICO

Es el recurso central de la presente guía, te proponemos que lo conozcas para que tengas un adecuado manejo de sus componentes.

Descripción

7A

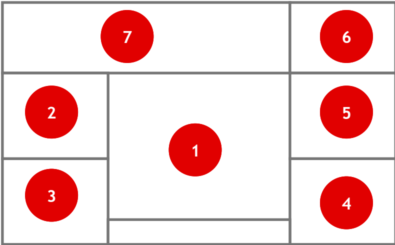
La Tienda de Robótica trae el **Kit Básico de Arduino** que comprende una cuidadosa selección de los mejores componentes para iniciar en el mundo de Arduino. Es una herramienta diseñada para los principiantes que cuenta con todo lo necesario para desarrollar programas que contribuyan al aprendizaje y manejo del Arduino.



Distribución del kit

7B

KIT BÁSICO



1

1 x Arduino Uno

1 x Protoboard

2

1 x Fotocelda

1 x Potenciómetro 10K

1 x Reed switch

2 x Pulsadores NO

3

5 x Led 5mm

1 x Led multicolor

1 x Piezo eléctrico

4

1 x Batería Alcalina 9V

1 x Broche batería

1 x Conector 2.1mm

5

5 x Resistencia 220Ω

5 x Resistencia 1KΩ

5 x Resistencia 10KΩ

2 x Transistor 2N3904

6

1 x Imán

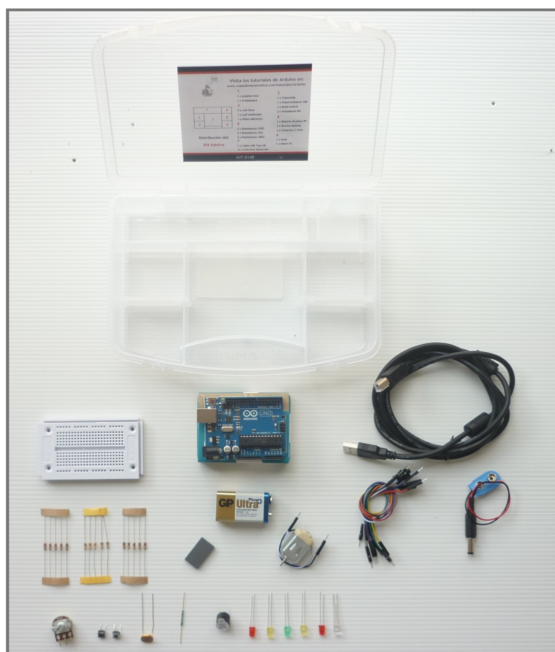
1 x Motor DC

7

1 x Cable USB Tipo AB

10 x Conectores rápidos MM

Fotos



DESPLIEGUE DE COMPONENTES



EN SU CAJA DE KIT

8. FRITZING

Fritzing es un software para diseñar los montajes en protoboard y a partir de ello generar el plano y la PCB (Circuito impreso), así de fácil :)

Software

8A

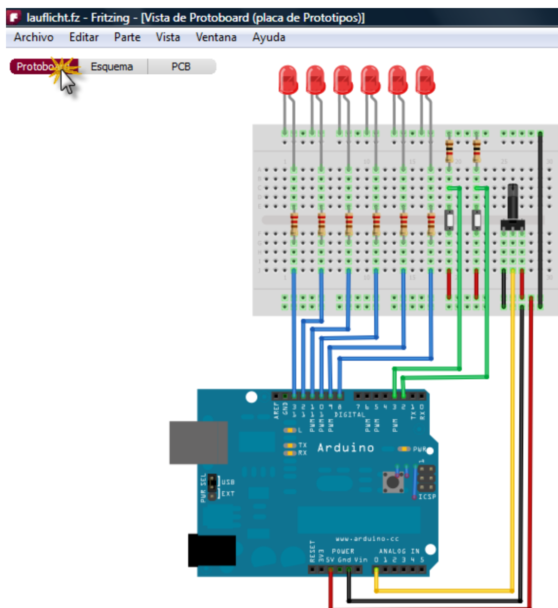


Fritzing permite a los diseñadores, artistas, investigadores y aficionados documentar sus prototipos basados en Arduino y crear esquemas de circuitos impresos para su posterior fabricación.



Vista protoboard

8B



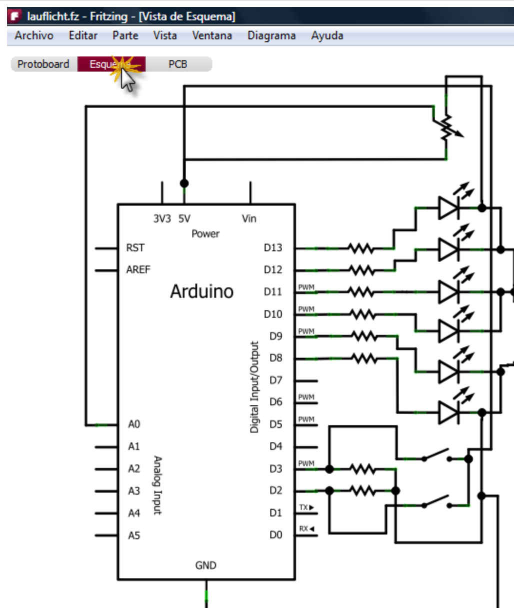
Ajuste de la velocidad de los LEDS

Realizado por tux60

Tomado de fritzing.org/projects/interruptgesteuertes-lauflicht/

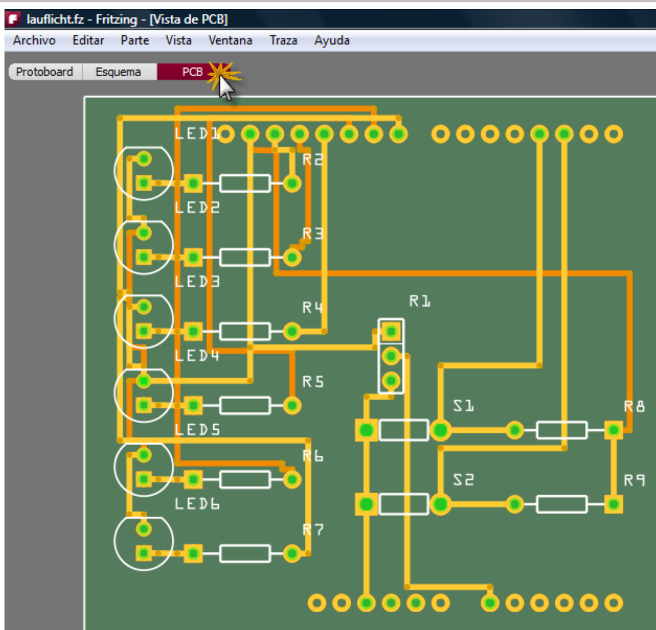
8C

Vista esquema



8D

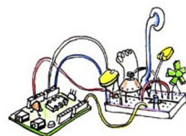
Vista PCB—Circuito impreso



EJERCICIO MÁSTER

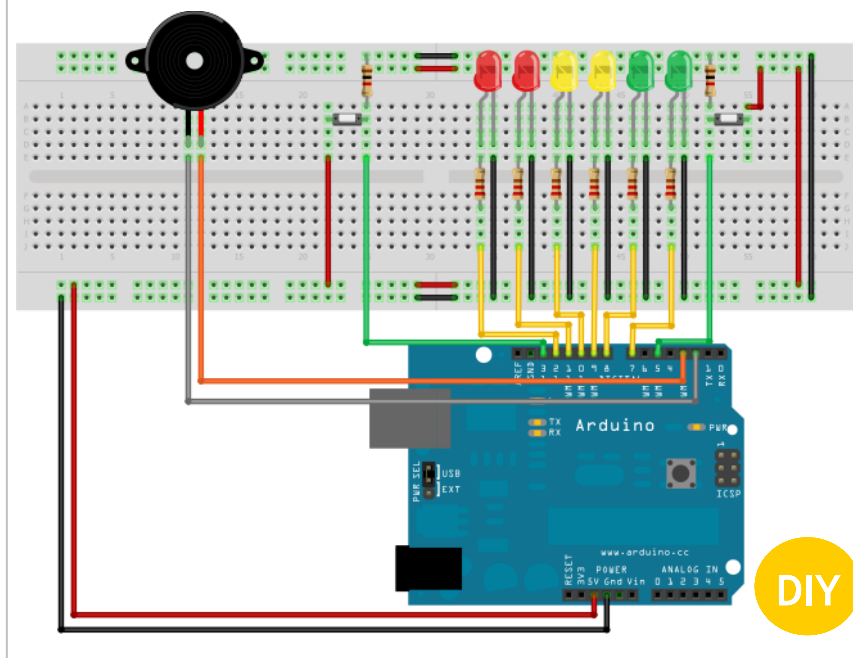
1

En el siguiente ejercicio queremos que afiances tus capacidades de montaje en Fritzing, es un ejercicio que requiere que coloques a prueba tus capacidades de reconocimiento de montajes electrónicos y cableado entre el Arduino y una protoboard, es un resumen práctico de los capítulos anteriores. Para desarrollarlo te recomendamos que trabajes junto con el software y de la imagen que se presenta a continuación. El software lo puedes encontrar en:



fritzing.org/download/

8E



DIY

9. TUTORIALES

Estos tutoriales son diseñados con ejercicios prácticos que permiten comprender el concepto y de manera constante se está verificando lo aprendido mediante preguntas. Finalmente se plantean ejercicios que involucran casos de la vida real.

T0 Conoce como son los tutoriales

- A Título del tutorial
- B Lo que debes saber y lo que vas aprender
- C Los materiales que necesitas
- D Más cerca de ti, contenido multimedia
- E Plano del montaje
- F Esquema del montaje

G Prueba tu aprendizaje

H Código de programación paso a paso

I Tips de mucha utilidad

CÓDIGO DE PROGRAMACIÓN

```
/*
 -------------------------------------
  Escritura serial
 -------------------------------------

  Consiste en escribir por la pantalla del computador (consola serial)
   una letra predeterminada, la primera vez que se escribe esta
   un LED se enciende, si se vuelve a escribir por segunda vez
   el LED se apaga.

 Cosas de Mecatrónica y Tienda de Robótica

*/

//--------------------------------------------------
//Declara puertos de entradas y salidas y variables
//--------------------------------------------------
int led = 13;              //Pin donde se encuentra el LED, salida
char leer;                 //Variable donde se almacena la letra
boolean prendido=false;  //Estado LED la primera vez, apagado

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  Serial.begin(9600);    //Inicia comunicación serial
  pinMode(led, OUTPUT);  //Configurar el LED como una salida
}

//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  //Guardar en una variable el valor de la consola serial
  leer=Serial.read();

  // Si es la letra 'a' y además el LED está apagado
  if ( (leer=='a') && (prendido==false) )
  {
      digitalWrite(led,HIGH);  // Enciende el LED
      prendido=true;           // Actualiza el estado del LED
  }
  // Si es la letra 'a' y además el LED está encendido
  else if ( (leer=='a') && (prendido==true) )
  {
      digitalWrite(led,LOW);   // Apaga el LED
      prendido=false;          // Actualiza el estado del
  }

}

//Fin programa
```

H

TIPS

1- Las tablas booleanas son muy útiles para entender la lógica de los programas, sabes cuál es la tabla booleana de AND(&&), OR(||) y NOT(!)

A AND B

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A OR B

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

NOT A

A	NOT A
0	1
1	0

I

80

EJERCICIOS

1 Avisos luminosos S.A. te ha contratado para que realices un programa que tiene la característica de recibir datos, porque la idea es que el aviso de luces se ubique en el piso 130 del Edificio Bulli y tu lo puedas controlar desde el piso 1, así que por cada letra que le escribas por teclado a la Consola serial el programa debe hacer determinada rutina con tres LED, si le escribes:

- Letra A: Prende los tres LED
- Letra B: Prende el LED1 por un segundo y lo apaga, luego prende el LED2 por un segundo y lo paga y finalmente prende el LED3 por un segundo y lo apaga [...] comenzar, esta rutina genera una sensación de movimiento
- Letra C: Apaga los tres LED

J

LED1 LED2 LED3

Arduino UNO

2 Tu abuelito tiene un cultivo de grandes y verdes lechugas, este cultivo tiene 3 aspersores de agua y tu abuelito quiere prender estos aspersores desde su computador solo escribiendo unos valores por teclado, los aspersores tienen un estado digital (Valor 0 es apagado y valor 1 es prendido). Debes realizar un programa que lea una cadena de tres valores, para este ejercicio los aspersores los vamos a simular con tres LED 1, 2 y 3 por ejemplo:

Si tu abuelito escribe 000 : Todos los aspersores de agua deben estar apagados

Si tu abuelito escribe 101 : El aspersor 1 y 3 se prenden pero el dos debe estar apagado

Si tu abuelito escribe 001 : El aspersor 1 y 2 se apagan pero el tres debe estar prendido

MI CUADERNO DE APUNTES

K

81

J Ejercicios aplicados

K Expresa tus ideas

T1 Hola mundo—LED intermitente

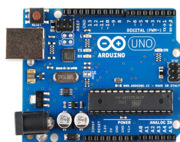
¿Qué aprendo?

- Activar una salida digital
- Encender un LED en ON/OFF
- Temporizar una señal de salida
- Sintaxis de un programa en Arduino

Conocimientos previos

- Señal digital
- Función digitalWrite()
- Polaridad de un LED
- Conexión de la placa Arduino al computador

Materiales

1 

Arduino UNO

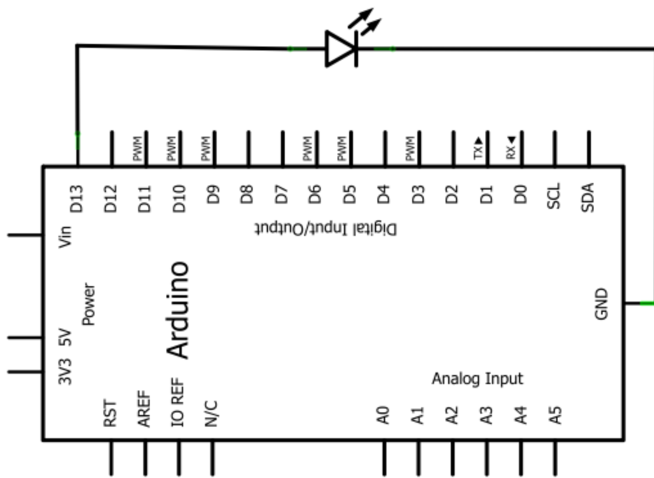
1 

LED Verde

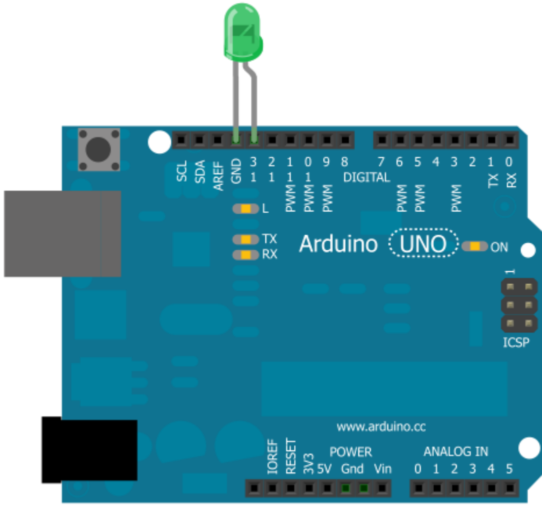
1 

Cable USB Tipo AB

PLANO



ESQUEMA



PREGUNTA

1— ¿Cuál es el valor de esta resistencia?



______Ω

______KΩ

2— ¿Qué hace está función?

digitalRead()

____

____

3— Completa

$$I = \frac{?}{R}$$

CÓDIGO DE PROGRAMACIÓN

```
/*
 -----------------------------------
  Hola Mundo
 -----------------------------------
 Enciende un LED por un segundo y lo apaga por el mismo tiempo
*/

//------------------------------------
//Función principal
//------------------------------------
void setup()     // Se ejecuta cada vez que el Arduino se inicia
{
   pinMode(13,OUTPUT); // Inicializa el pin 13 como una salida
}


//------------------------------------
//Función cíclica
//------------------------------------
void loop()      // Esta función se mantiene ejecutando
{                // cuando este energizado el Arduino
  digitalWrite(13,HIGH);  // Enciende el LED
  delay(1000);            // Temporiza un segundo (1s = 1000ms)
  digitalWrite(13,LOW);   // Apaga el LED
  delay(1000);            // Temporiza un segundo (1s = 1000ms)
}

// Fin del programa
```

TIPS

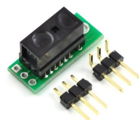
1- El // en programación se utiliza para hacer comentarios, es muy útil para que puedas explicar algo acerca de la sintaxis de una línea de código. Un ejemplo de su uso:

```
digitalWrite(13,LOW);   // Apaga el LED
```

2- Las señales digitales (Encendido o apagado) están muy presentes en todos los sistemas, y muchos sensores trabajan sobre este principio, te invitamos a conocer algunos:



Sensor PIR
Detecta movimiento



Sensor óptico SHARP
Detecta la presencia de algún objeto en un rango de 5cm

EJERCICIOS

1

A partir del código de programación explicado anteriormente, queremos que un LED prenda de manera intermitente de la siguiente manera:

A) 3 segundos prendido y 3 segundos apagado

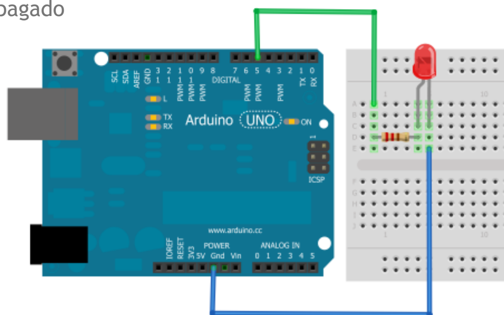
B) 200 milisegundos prendido y 500 milisegundos apagado

Tip: 1s = 1000ms

2

La empresa de automatización NRJ Inc. te contrata para hacer un montaje de un LED Intermitente en una placa Arduino UNO, con el único requisito de que el LED debe estar ubicado en el Pin 5, ellos muy amablemente te han facilitado el esquema, tu tarea es:

A) Realizar el montaje y la respectiva programación de 2 segundos prendido y 1 segundo apagado



3

Un Strober es un bombillo que prende y apaga muy rápido, muy usado en las fiestas, tu misión es realizar tu strober casero con un LED, con el mismo esquema montado en este tutorial.



MI CUADERNO DE APUNTES

T2 Encender un LED con un pulsador

¿Qué aprendo?

- Cablear un circuito
- Condicional If/else
- Estado de un pulsador
- Leer una entrada digital y escribir una salida digital

Conocimientos previos

- Señal digital
- Función digitalWrite() y digitalRead()
- Divisor de voltaje
- Condicional y operadores de comparación

Materiales

1 

Arduino UNO

1 

LED Amarillo

1 

Protoboard

1 

Pulsador

1 

Cable USB Tipo AB

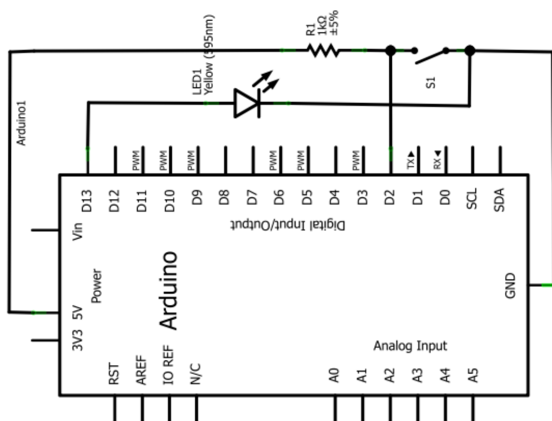
1 

Resistencia 1K

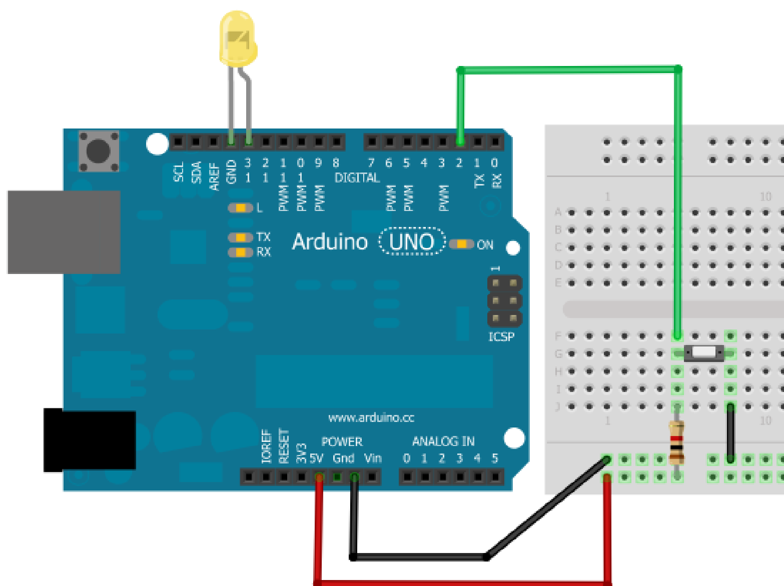
4 

Conectores MM

PLANO



ESQUEMA

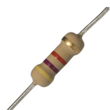


PREGUNTA

1— ¿Cuál es el valor de esta resistencia?

______Ω

______KΩ



2— ¿Qué hace está función?

digitalWrite()

____

____

3— Un ejemplo de un lenguaje de alto nivel

____

CÓDIGO DE PROGRAMACIÓN

```
/*
 -----------------------------------
  Encender LED con un pulsador
 -----------------------------------

 Oprimir un pulsador y mientras este se mantenga accionado
  un LED se enciende

 Cosas de Mecatrónica y Tienda de Robótica

*/


//------------------------------------
//Declara puertos de entradas y salidas
//------------------------------------
int pulsador=2;          //Pin donde se encuentra el pulsador, entrada
int led=13;              //Pin donde se encuentra el LED, salida

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
 pinMode(pulsador, INPUT); //Configurar el pulsador como una entrada
 pinMode(led,OUTPUT);      //Configurar el LED como una salida
}

//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

   //Condicional para saber estado del pulsador
   if (digitalRead(pulsador)==HIGH)
   {
     //Pulsador oprimido
     digitalWrite(led,HIGH); //Enciende el LED
   }
   else
   {
     //Pulsador NO oprimido
     digitalWrite(led,LOW);  //Apaga el LED
   }
}
//Fin programa
```

TIPS

1- Cuando estés programando en el Software de Arduino, muchas cosas de las que escribes son palabras reservadas por el lenguaje, todas las palabras reservadas las puedes encontrar en la sección **5S**, al escribirlas éstas se colocan en un color diferente, este es un truco para saber que esta bien, por ejemplo:

```
void loop() {
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

2- Todas la instrucciones de programación para Arduino, se encuentran totalmente documentadas con claros ejemplos de cómo se utilizan, te invitamos a que visites: **arduino.cc/en/Reference/HomePage**

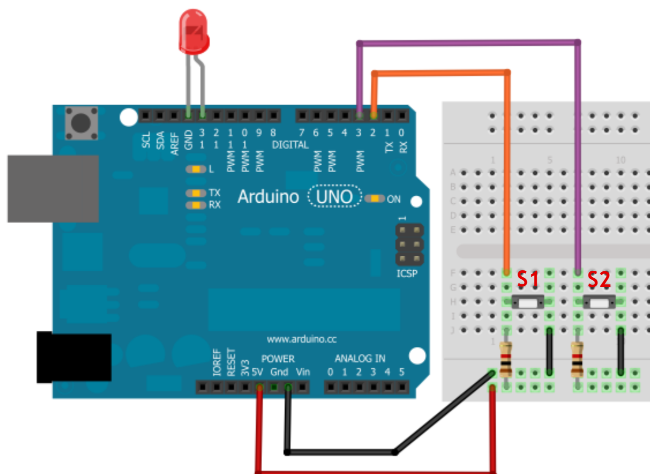


EJERCICIOS

1

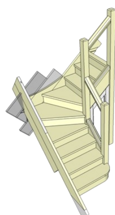


La multinacional francesa **Flante** experta en maquinaria industrial te ha contratado para que automatices una máquina cortadora de papel. La condición principal es que el operario de la máquina cuando vaya a realizar el corte siempre mantenga las dos manos ocupadas, esta es una regla de seguridad industrial para evitar accidentes. El operario debe oprimir los dos pulsadores uno con cada mano y la cuchilla cortadora debe bajar y hacer el corte. El siguiente montaje simula el control de la máquina, los dos pulsadores (S1 y S2) y el LED rojo simula la cuchilla cortadora.



2

Tu padre quiere que realices un sistema de iluminación LED para las escaleras de la casa. La condición es que si estás arriba y pulsas a S1 o si estás abajo y pulsas S2 el LED Rojo se enciende y al dejar de pulsar se apaga. Como guía de montaje toma la imagen anterior.



MI CUADERNO DE APUNTES

T3 Lectura serial de una entrada digital

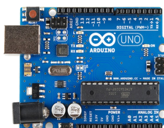
¿Qué aprendo?

- Manejar una entrada digital
- Ver datos por la pantalla del computador
- Consola serial
- Leer una entrada digital y escribir por consola serial

Conocimientos previos

- Señal digital
- Función digitalRead() y Serial.println()
- Opción de Consola serial, ver 6F (paso 3)

Materiales

1 
Arduino UNO

1 
Pulsador

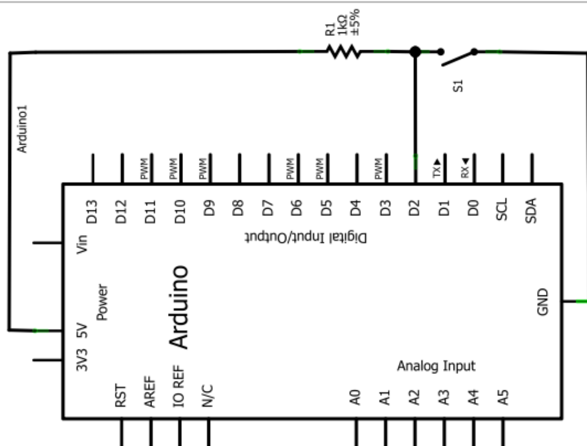
1 
Protoboard

1 
Cable USB Tipo AB

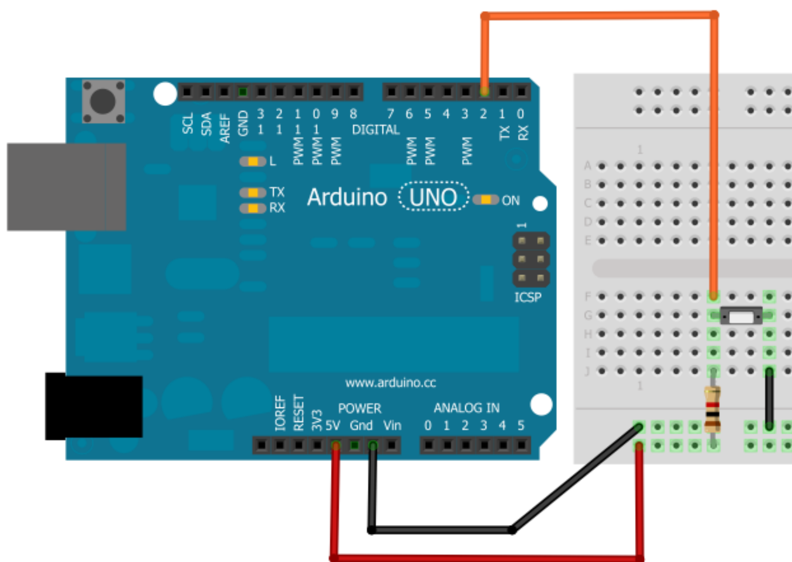
1 
Resistencia 1K

4 
Conectores MM

PLANO



ESQUEMA

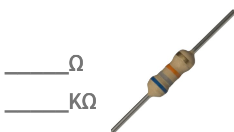


PREGUNTA

1— ¿Cuál es el valor de esta resistencia?

______Ω

______KΩ



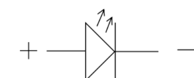
2— ¿Qué hace está función?

Serial.println()

____

____

3— Este símbolo a que corresponde



____

CÓDIGO DE PROGRAMACIÓN

```
/*
 ------------------------------------
  Lectura serial de una entrada digital
 ------------------------------------

 Leer una entrada digital y mostrar por la pantalla del
  computador (consola serial) el estado del pulsador
  cuando es oprimido

 Cosas de Mecatrónica y Tienda de Robótica

*/


//------------------------------------
//Declara puertos de entradas y salidas
//------------------------------------
int boton=2;        //Pin donde se encuentra el pulsador, entrada

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  //Configuración
  pinMode(boton,INPUT);  //Configurar el boton como una entrada
  Serial.begin(9600);    //Inicia comunicación serial
}


//------------------------------------
//Funcion ciclica
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

   //Guardar en una variable entera el valor del boton 0 ó 1
   int estado = digitalRead(boton);

   //Condicional para saber estado del pulsador
   if (estado==1)
   {
     // Pulsado
     Serial.println("Pulsado");      //Imprime en la consola serial
   }                                 //  "Pulsado"
   else
   {
     // No esta pulsado
     Serial.println("NO Pulsado"); //Imprime en la consola serial
   }                               // "NO Pulsado"

   delay(100); //Retardo para la visualización de datos en la consola

}

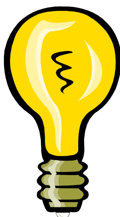
//Fin programa
```

TIPS

1- La codificación binaria es muy importante para transmitir datos entre dispositivos, son las largas cadenas de 0 y 1, por ejemplo 00011101010101 esto podría ser un mensaje que contiene información referente a una clave personal para acceder a un edificio. Los números en base 10 se pueden representar como valores binarios:

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

2- Para recordar



- Para leer una señal digital usa: **digitalRead(numeroPin);**

- Para escribir una señal digital usa: **digitalWrite(numeroPin, valor);**

- Una salida o entrada digital siempre es **HIGH** o **LOW**

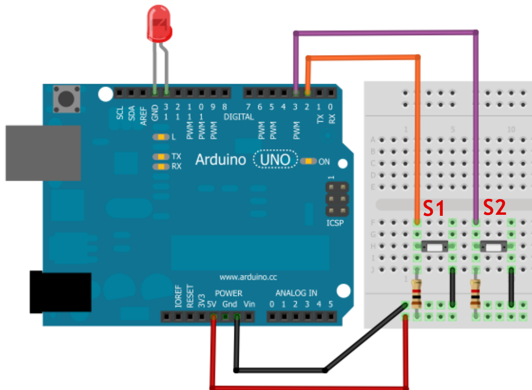
EJERCICIOS

1

Una empresa de gaseosas tiene un sistema con dos sensores, uno de ellos indica si la botella se ha llenado con el líquido y el otro sensor indica si la botella ya tiene la tapa. Para este caso simularemos los dos sensores por medio de dos pulsadores (S1 y S2).



La rutina se describe de esta manera: si la botella se llena de manera adecuada (se debe activar S1 y mostrar por consola "Gaseosa llena") luego de ello si tiene la tapa colocada (se debe activar S2 y mostrar por consola "Gaseosa tapada"), al finalizar el proceso se debe encender un LED que indica que el proceso terminó bien y además se debe mostrar un mensaje por la consola "Gaseosa bien empacada". Recuerda que primero se debe activar S1 y luego S2 para que le proceso sea válido.



2

Unas luces navideñas modernas son las siguientes: mientras se mantenga pulsado S1 una variable entera inicializada en 0 se comienza a incrementar de 20 en 20, al soltar S1 la variable para de incrementarse, cuando se pulse S2 el valor de la variable se debe cargar a un "Hola Mundo" del LED, esto quiere decir que el LED va estar intermitente en intervalos de tiempo iguales al valor de la variable. Por consola serial debes ir mostrando el valor de la variable. Para que puedas volver la variable a 0 y puedas hacer otra rutina de intermitencia, coloca la condición de que S1 y S2 se opriman al tiempo, con ello la variable entera debe volver a 0.



MI CUADERNO DE APUNTES

T4 Lectura serial de una entrada análoga

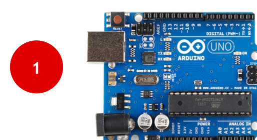
¿Qué aprendo?

- Manejar una entrada análoga
- Ver datos por la pantalla del computador
- Múltiples estados de un potenciómetro
- Leer una entrada análoga

Conocimientos previos

- Señal análoga
- Función analogRead() y Serial.println()
- Opción de Consola serial, ver 6F (paso 3)

Materiales

1 

Arduino UNO

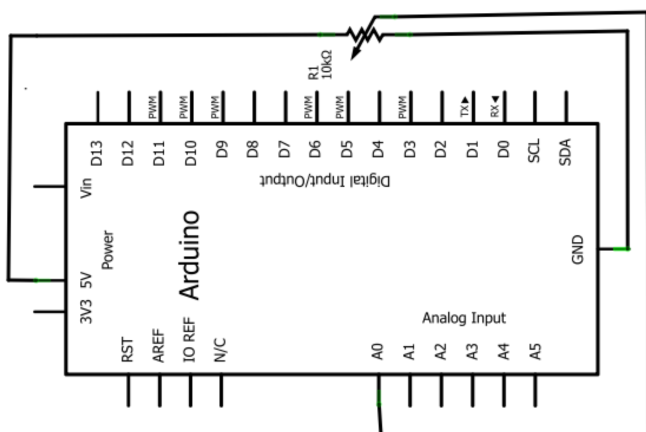
1 
Potenciómetro 10K

1 
Protoboard

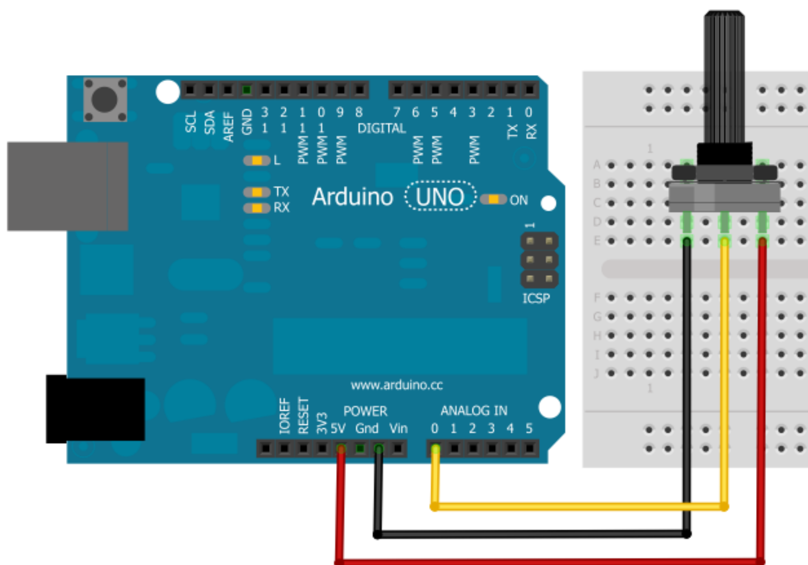
1 
Cable USB Tipo AB

3 
Conectores MM

PLANO

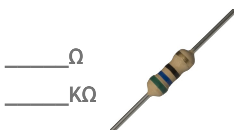


ESQUEMA



PREGUNTA

1– ¿Cuál es el valor de esta resistencia?



______Ω

______KΩ

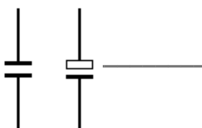
2– ¿Qué hace está función?

analogRead()

____

____

3– Este símbolo a que corresponde



____

CÓDIGO DE PROGRAMACIÓN

```
/*
 -------------------------------------
  Lectura serial de entrada análoga
 -------------------------------------

  Leer una entrada análoga y mostrar por la pantalla del
   computador (consola serial) el valor luego de girar
   el potenciómetro

 Cosas de Mecatrónica y Tienda de Robótica

*/

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  Serial.begin(9600); //Inicia comunicación serial
}

//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  //Guardar en una variable entera el valor del potenciómetro 0 a 1024
  int valor= analogRead(A0);

  //Imprime en la consola serial el valor de la variable
  Serial.println(valor);

  //Retardo para la visualización de datos en la consola
  delay(100);

}

//Fin programa
```

TIPS

1- Te invitamos a que conozcas algunos tipos de potenciómetros



SoftPot
Sistema touch

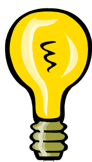


Trimmer
Alta precisión



Encoder RGB
Giro continuo

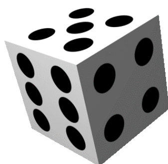
2- Para recordar



- Para leer una señal análoga usa: **analogRead(numeroPin);**
- Para escribir una señal análoga de PWM usa: **analogWrite(numeroPin, valor);**
- Una entrada análoga va de **0** o **1023**
- Una salida análoga de PWM va de **0** o **255**

EJERCICIOS

1

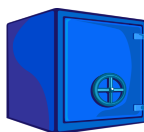


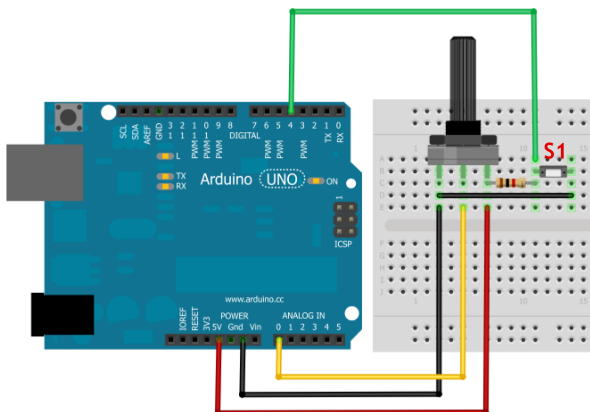
Este reto es sencillo, pero te va a dar una idea de cómo hacer grandes cosas escribiendo unas pocas líneas de código. Como bien lo sabes, la lectura de una señal análoga te da un valor de 0 a 1023 (si tienes alguna duda solo revisa el código de la página anterior).

El desafío de este ejercicio consiste en mostrar por la consola serial un número entre 0 y 10, este número debe cambiar cuando muevas el potenciómetro. El montaje de este ejercicio usa el mismo Esquema de conexiones que el expuesto en la página 75. Sugerencia... Revisa la función map()

2

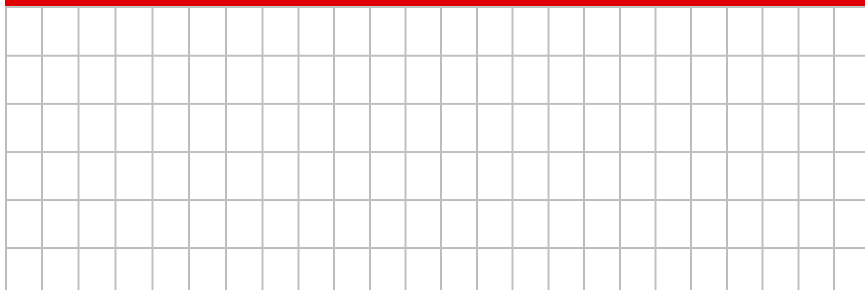
Piensa en un número secreto de 0 a 1023 y guárdalo en una variable entera, haz un programa para que tu amigo deba mover el potenciómetro, si el número que el va generando (mostrar por consola serial el valor de la lectura del potenciómetro) al girar el potenciómetro está 10 números por arriba o 10 números por abajo al número que tu pensaste y confirma con el pulsador S1 que ese es el número, el programa debe mostrar por consola "Adivinaste el número" de lo contrario "Intenta de nuevo".





Este caso podría ser un ejemplo para que lo apliques a una Caja de seguridad, para que guardes mucho dinero ;)

MI CUADERNO DE APUNTES



T5

Escritura serial

¿Qué aprendo?

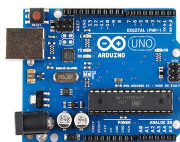
- Entrada por consola (teclado)
- Variables booleanas
- Estado de un LED
- Escritura serial digital

Conocimientos previos

- Señal digital
- Función digitalWrite() y Serial.read()
- Configuración de una comunicación serial
- Polaridad de un LED

Materiales

1



Arduino UNO

1



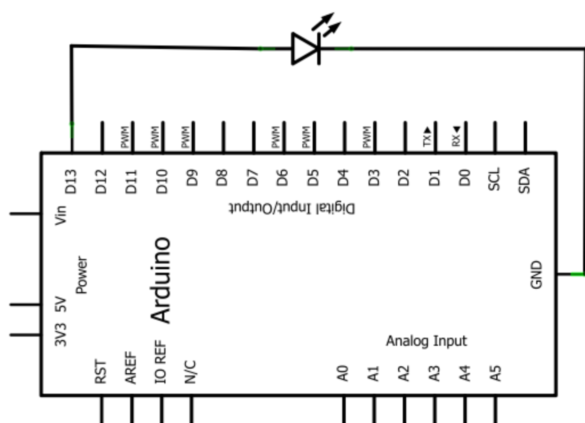
LED Verde

1

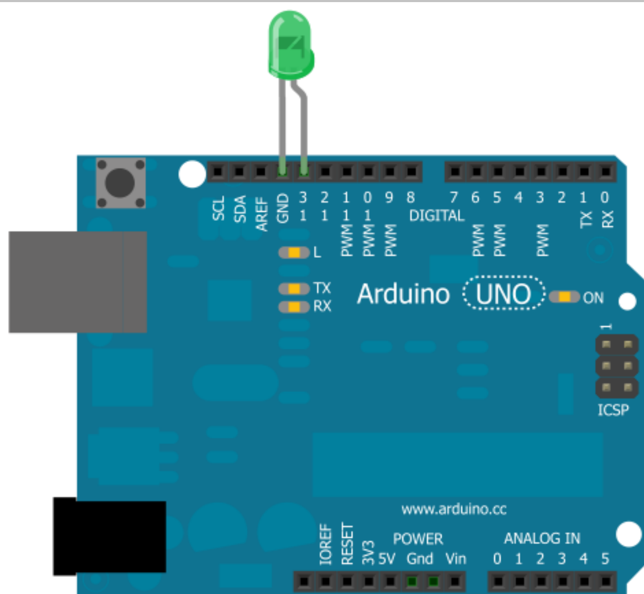


Cable USB Tipo AB

PLANO



ESQUEMA

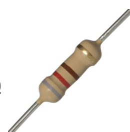


PREGUNTA

1— ¿Cuál es el valor de esta resistencia?

______Ω

______KΩ



2— ¿Qué hace está función?

Serial.read()

____

____

3— Nombra 4 tipos de variables

a.________

b.________

c.________

d.________



CÓDIGO DE PROGRAMACIÓN

```
/*
 -------------------------------------
  Escritura serial
 -------------------------------------

  Consiste en escribir por la pantalla del computador (consola serial)
   una letra predeterminada, la primera vez que se escriba está
   un LED se enciende, si se vuelve a escribir por segunda vez
   el LED se apaga.

 Cosas de Mecatrónica y Tienda de Robótica

*/


//------------------------------------------------
//Declara puertos de entradas y salidas y variables
//------------------------------------------------
int led = 13;            //Pin donde se encuentra el LED, salida
char leer;               //Variable donde se almacena la letra
boolean prendido=false; //Estado LED la primera vez, apagado


//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  Serial.begin(9600);    //Inicia comunicación serial
  pinMode(led, OUTPUT);  //Configurar el LED como una salida
}


//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  //Guardar en una variable el valor de la consola serial
  leer=Serial.read();

  // Si es la letra 'a' y además el LED está apagado
  if ( (leer=='a') && (prendido==false) )
  {
      digitalWrite(led,HIGH);  // Enciende el LED
      prendido=true;           // Actualiza el estado del LED
  }
  // Si es la letra 'a' y además el LED está encendido
  else if ( (leer=='a') && (prendido==true) )
  {
      digitalWrite(led,LOW);   // Apaga el LED
      prendido=false;          // Actualiza el estado del LED
  }

}

//Fin programa
```

TIPS

1- Las tablas booleanas son muy útiles para entender la lógica de los programas, ¿sabes cuál es la tabla booleana de **AND(&&)**, **OR(||)** y **NOT(!)** ?

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

A	NOT A
0	1
1	0

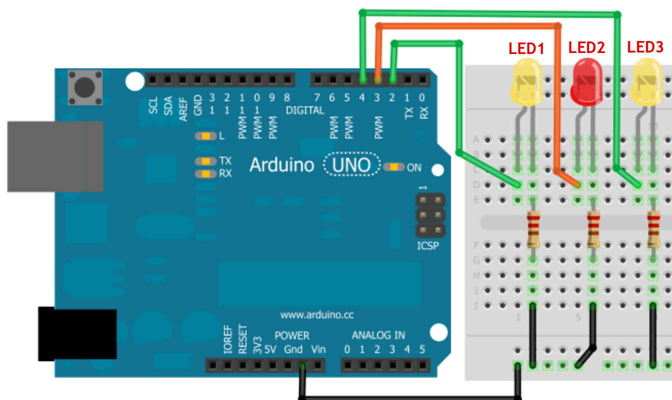
EJERCICIOS

1

Avisos luminosos S.A. te ha contratado para que realices un programa que tiene la característica de recibir datos, porque la idea es que el aviso de luces se ubique en el piso 130 del Edificio Bulli y tu lo puedas controlar desde el piso 1, así que por cada letra que le escribas por teclado a la Consola serial el programa debe hacer determinada rutina con tres LEDs, si le escribes:



- **-Letra A:** Prende los tres LED
- **-Letra B:** Prende el LED1 por un segundo y lo apaga, luego prende el LED2 por un segundo y lo apaga y finalmente prende el LED3 por un segundo y lo apaga, y vuelve a comenzar. Esta rutina genera una sensación de movimiento
- **-Letra C:** Apaga los tres LED



2



Tu abuelito tiene un cultivo de grandes y verdes lechugas, este cultivo tiene 3 aspersores de agua y tu abuelito quiere prender estos aspersores desde su computador solo escribiendo unos valores por teclado, los aspersores tienen un estado digital (Valor 0 es apagado y valor 1 es prendido). Debes realizar un programa que lea una cadena de tres valores, para este ejercicio los aspersores los vamos a simular con tres LED 1, 2 y 3 por ejemplo:

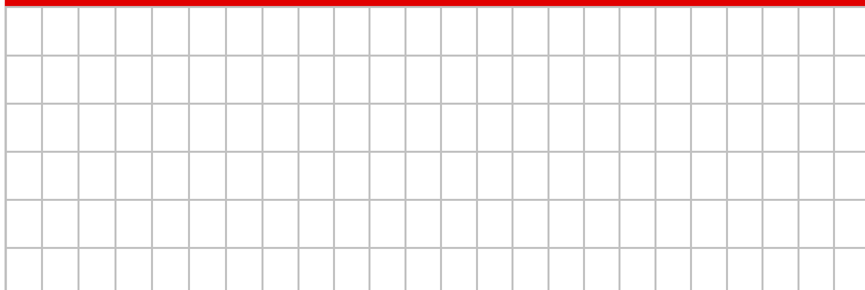


Si tu abuelito escribe **000** : Todos los aspersores de agua deben estar apagados

Si tu abuelito escribe **101** : El aspersor 1 y 3 se prenden pero el dos debe estar apagado

Si tu abuelito escribe **001** : El aspersor 1 y 2 se apagan pero el tres debe estar prendido

MI CUADERNO DE APUNTES



T6

Encender un LED por PWM

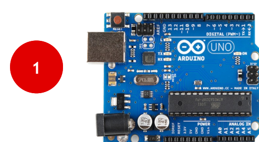
¿Qué aprendo?

- Encender un LED de manera proporcional
- Apropiar el concepto de PWM
- Escribir una salida análoga
- If/else con operadores lógicos

Conocimientos previos

- PWM
- Función analogWrite()
- Polaridad de un LED
- Incrementar y manipular variables

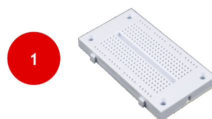
Materiales

1 

Arduino UNO

1 

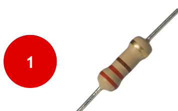
LED Amarillo

1 

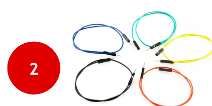
Protoboard

1 

Cable USB Tipo AB

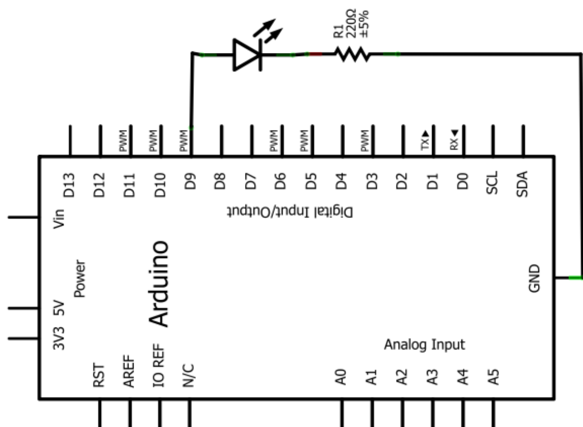
1 

Resistencia 220Ω

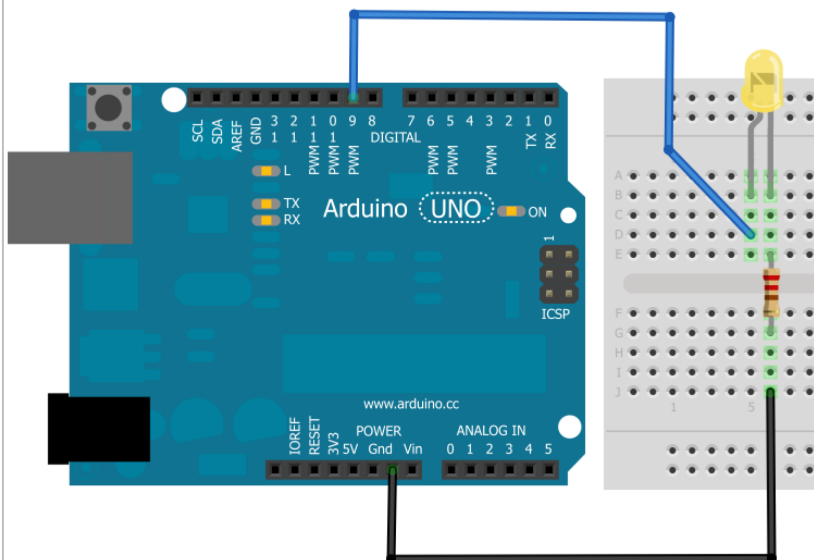
2 

Conectores MM

PLANO



ESQUEMA

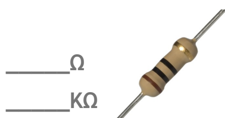


PREGUNTA

1— ¿Cuál es el valor de esta resistencia?

______Ω

______KΩ



2— ¿Qué hace está función?

analogWrite()

____

____

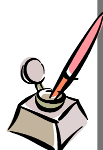
3— A que corres-ponden estos ope-radores

&&_________

||_________

==_________

!=_________



CÓDIGO DE PROGRAMACIÓN

```
/*
 -------------------------------------------
  Enciende/Apaga un LED de forma proporcional
 -------------------------------------------

  Programa que enciende proporcionalmente un LED cuando
   llega a su máximo punto de brillo comienza a apagarse
   proporcionalmente.

 Cosas de Mecatrónica y Tienda de Robótica

*/


//------------------------------------------------
//Declara puertos de entradas y salidas y variables
//------------------------------------------------
int brillo = 0;         //Variable de brillo inicia en 0
int variacion = 5;      //Variable de incremento configurada de 5 en 5
int led = 9;            //Pin donde se encuentra el LED, salida


//------------------------------------
//Funcion principal
//------------------------------------
void setup () // Se ejecuta cada vez que el Arduino se inicia
{
  pinMode(led, OUTPUT); //Configurar el LED como una salida
}


//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop () // Esta funcion se mantiene ejecutando
{            //  cuando este energizado el Arduino

  // Escritura analoga (PWM) en el LED escribo el valor de brillo
  analogWrite(led, brillo);

  // Incremento la variable brillo de 5 en 5
  brillo = brillo + variacion;

  // Nota: PWM ---->  0 - 255

  // Si el brillo es 0 o 255
  if (brillo == 0 || brillo == 255)
    variacion = -variacion;  //La variación se vuelve negativa

  delay (30); //Tiempo de incremento en el brillo

}


//Fin programa
```

TIPS

1- Estos elementos reciben señales de PWM y sirven para:



Bomba de agua

Variar la velocidad de bombeo

Micromotor

Variar la velocidad de giro







LED

Variar la intensidad de luz

Servomotor

Variar la posición en grados



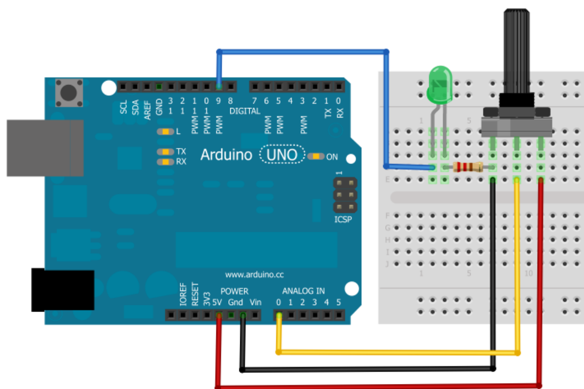
EJERCICIOS

1



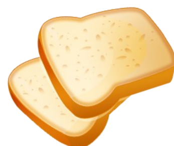
Tu mamá quiere tener unas luces de intensidad variables en la sala de tu casa, quiere poca luz para los momentos en que la familia comparte para mantener la armonía pero requiere bastante luz cuando realiza sus tejidos, ella quiere tener el control de la intensidad de luz mediante un potenciómetro y que la bombilla LED se ilumine de acuerdo a lo graduado por ella. Tu tarea es realizar el programa que al leer una entrada análoga de un potenciómetro (0—1023) se ajuste el valor de PWM (0-255) de un LED, recuerda escalar el valor… map()





2

Un novedoso horno para pan está próximo a salir al mercado. Este horno es distinto a todos los demás porque le puedes ajustar la temperatura y el tiempo de horneado desde el computador. Tu tarea es realizar un programa que lee por Consola Serial la temperatura en °C (0-255) luego se separa por una coma (,) y a continuación viene el tiempo de horneado en segundos (1-10). El valor de la temperatura va a ser el PWM que se le va a ajustar al LED y el tiempo de horneado es el tiempo durante el cual el LED se mantiene encendido a esa temperatura. Usa el Esquema de la pág. 83 como montaje, por ejemplo:

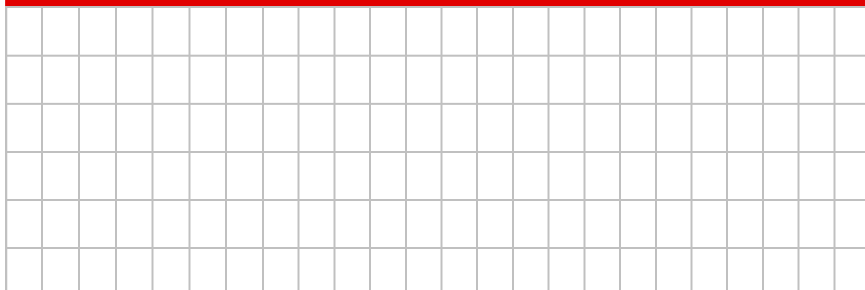


180,3 El LED se prende con un PWM de 180 durante 3 segundos

50,2 El LED se prende con un PWM de 50 durante 2 segundos

9,5 El LED se prende con un PWM de 9 durante 9 segundos

MI CUADERNO DE APUNTES



T7 Control ON/OFF con potenciómetro

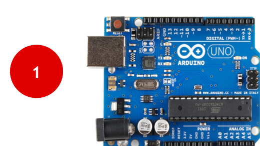
¿Qué aprendo?

- Salida digital
- Control ON/OFF
- Comparación
- Condicional a partir de un valor entero de una entrada análogo

Conocimientos previos

- If/else
- Función digitalWrite() y analogRead()
- Valor de una entrada análoga
- Condicional y operadores de comparación

Materiales

1


Arduino UNO

1

LED Verde

1

Protoboard

1

Potenciómetro 10K

1

Cable USB Tipo AB

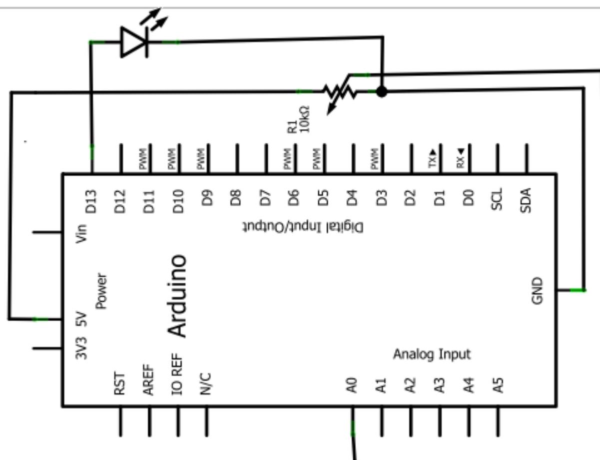
1

Resistencia 220Ω

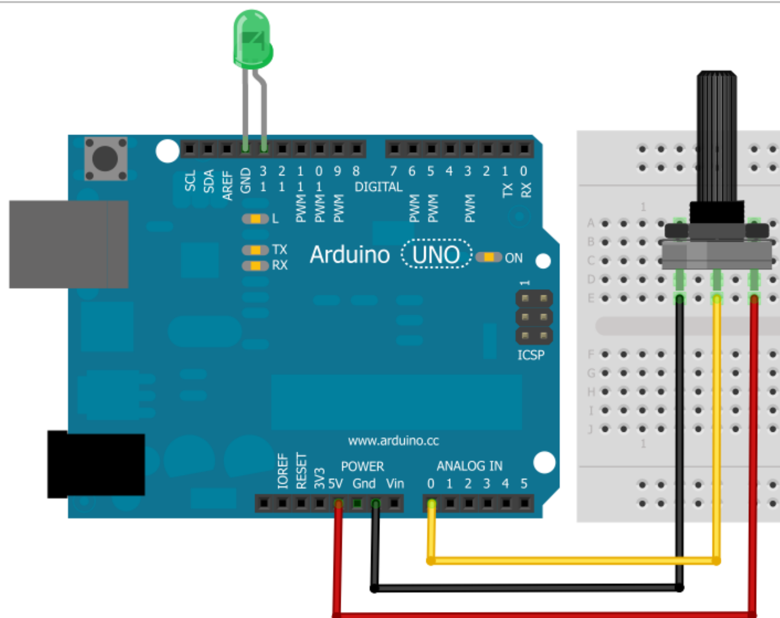
3

Conectores MM

P L A N O

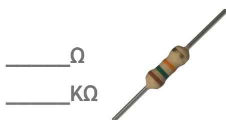


E S Q U E M A

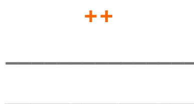


P R E G U N T A

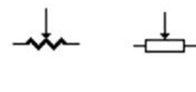
1— ¿Cuál es el valor de esta resistencia?



2— ¿Para que sirve el operador?



3— Este símbolo a que corresponde



CÓDIGO DE PROGRAMACIÓN

```
/*
 -----------------------------------------------
  Control ON/OFF con potenciómetro
 -----------------------------------------------

  Programa que enciende un LED cuando el valor de la entrada
   análoga comandada por el potenciómetro esta en cierto valor,
   cuando este valor cambia el LED se apaga, es un sistema con
   control ON/OFF

 Cosas de Mecatrónica y Tienda de Robótica

*/

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  Serial.begin(9600);  //Inicia comunicación serial
  pinMode(13,OUTPUT);  //Configurar el pin 13 como una salida
}

//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  //Guardar en una variable el valor de la lectura análoga
  int valor = analogRead(A0);

  Serial.println(valor);  //Imprime el valor por la consola

  //Si el valor es mayor o igual a 500
  if (valor >= 500)
  {
    digitalWrite(13,HIGH); //Enciende el LED en el pin 13
  }
  //Si el valor es menor a 500
  else
  {
    digitalWrite(13,LOW); //Apaga el LED en el pin 13
  }

  delay(100); //Retardo de 100ms para ver los datos de la consola
}

//Fin programa
```

TIPS



1- Debes tener mucha precaución al momento en que tu placa Arduino este energizada, si miras la palca por la parte inferior esta tiene todos sus puntos de soldadura al aire, si tienes la placa alimentada y la colocas encima de una superficie metálica, por ejemplo una mesa, es posible que la placa se dañe ya que queda en corto. Si vas a trabajar la placa Arduino te recomendamos trabajar sobre superficies de materiales aislantes como los son los sintéticos, madera o vidrio. Este es un consejo para que cuides tu placa.

EJERCICIOS

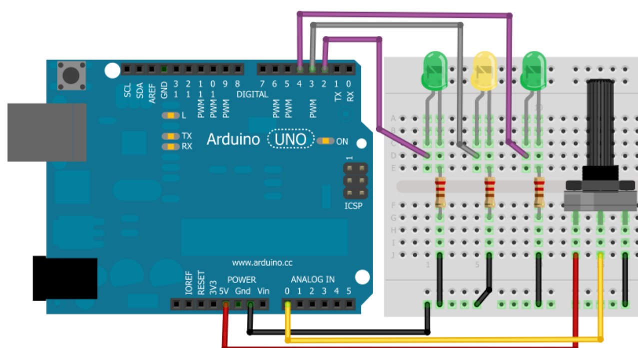
1 Un carro de tunnig tiene luces de neón en la parte de abajo, éste es la sensación de la feria. El piloto controla las luces (tres LEDs) por medio de un potenciómetro al lado de la palanca de cambios, él tiene tres opciones de luces de acuerdo al valor de lectura del potenciómetro que va de 0 a 1023:



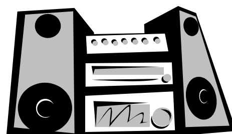
0 - 350 Intermitencia de los 3 LED a 100ms

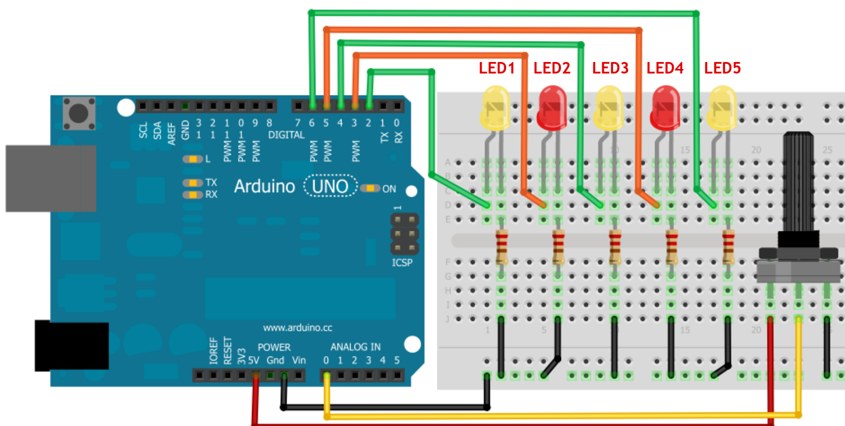
351 - 700 Intermitencia de los 3 LED a 500ms

701 - 1023 Intermitencia de los 3 LED a 1000ms



2 El equipo de sonido tiene una perilla para que le subas el volumen y además cuenta con 5 indicadores luminosos (LED) dispuestos uno tras otro en una línea recta (así como lo ves en el montaje) en la medida que el valor del potenciómetro va aumentando al girarlo, se va encendiendo el LED1, luego el LED2, luego el LED3 y así hasta el LED5. Si llegas al valor de 1023 todos los LED deberían estar prendidos y si comienzas a girar el potenciómetro en sentido contrario se van apagando los LED uno tras otro en el sentido inverso en que se encendieron, al final si estás en el valor 0 todos los LED deben estar apagados. Generar una sensación de movimiento.





T8 Control de un LED con una fotocelda

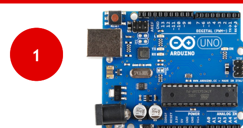
¿Qué aprendo?

- Escritura por PWM en un LED
- Leer una entrada análoga por medio de una fotocelda
- Trabajar con una variable
- Ajustar una entrada análoga a una salida análoga

Conocimientos previos

- Señal análoga
- Función analogWrite() y analogRead()
- PWM
- Imprimir datos por consola serial

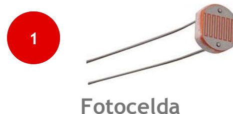
Materiales

1 
Arduino UNO

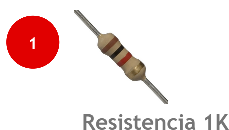
1 
LED Verde

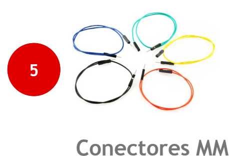
1 
Protoboard

1 
Cable USB Tipo AB

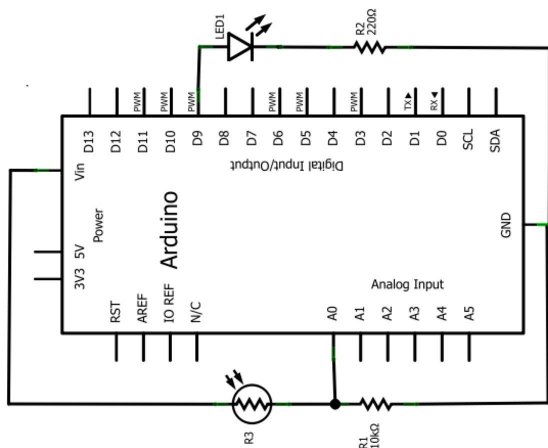
1 
Fotocelda

1 
Resistencia 10K

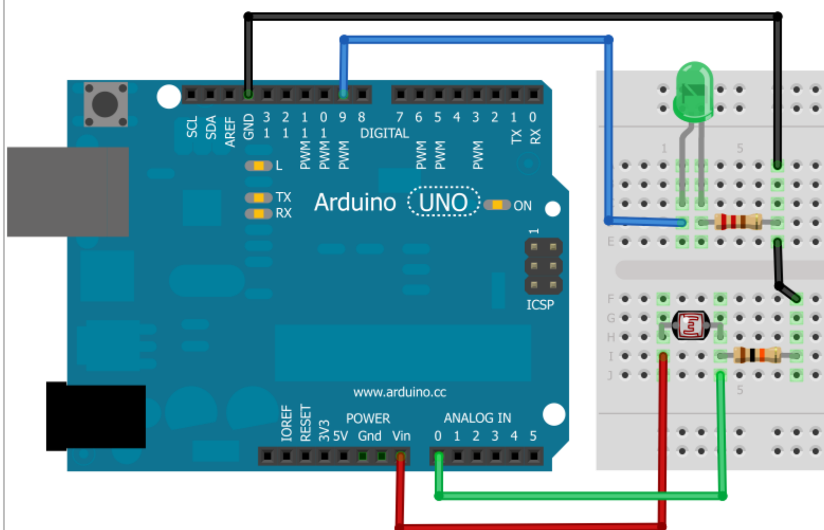
1 
Resistencia 1K

5 
Conectores MM

PLANO



ESQUEMA

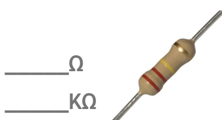


PREGUNTA

1— ¿Cuál es el valor de esta resistencia?

______Ω

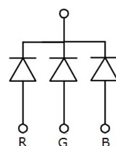
______KΩ



2— La letra A se guarda en una variable de tipo:

____

3— Este símbolo a que corresponde



____

CÓDIGO DE PROGRAMACIÓN

```
/*
 ---------------------------------------------
  Control de intensidad de un LED
 ---------------------------------------------

  Programa que enciende un LED de manera proporcional de
   acuerdo a la cantidad de luz que incida en una
   fotocelda.

 Cosas de Mecatrónica y Tienda de Robótica

*/
//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  Serial.begin(9600);  //Inicia comunicación serial
  pinMode(9,OUTPUT);   //Configurar el pin 9 como una salida de PWM
}

//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  //Guardar en una variable el valor de la lectura análoga de la
  // fotocelda
  int foto = analogRead(A0);

  //Verifica el valor máximo y realizar una conversión
  int conversion = 780 - foto;

  //Condicional para establecer un valor absoluto
  if ( conversion < 0)
    conversion = conversion * -1; //Multiplicar por -1 porque es negativo

  //Imprimir datos del valor de la lectura análoga de la fotocelda
  Serial.print("Foto : ");
  Serial.print(foto);
  Serial.println("");

  //Imprimir datos del valor de la conversión
  Serial.print("Conv : ");
  Serial.print(conversion);
  Serial.println("");

  //Escritura análoga de PWM en el LED de acuerdo a la conversión
  analogWrite(9, conversion);

  delay(100);  //Retardo para datos en la consola

}

//Fin programa
```

TIPS

1- Piensa por un momento en dónde puedes encontrar una fotocelda, éstos son algunos ejemplos:



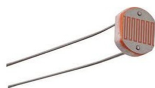
Calculadora

Cargar batería y apagado automático

Cámara digital

Verifica si hay necesidad de foto con flash







Medidor de luz

Mide una cantidad de luz y lo da en lux (lx)

Alumbrado público

Enciende la luz cuando llega la oscuridad

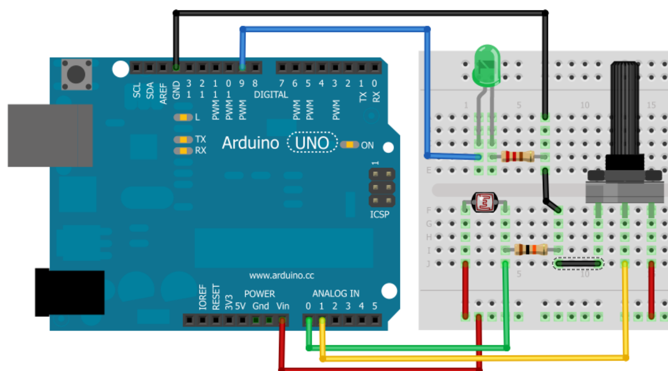


EJERCICIOS

1



Te proponemos que crees un sensor para medir la luz que irradia un cuerpo luminoso. Vamos a tener dos equipos el A y el B. En el equipo A va estar un LED que varia su brillo de acuerdo a la posición de un potenciómetro (Ver T6) y en el equipo B va estar comandado por una fotocelda. La idea es que dobles el LED y la fotocelda a 90°, los coloques lo más junto posible y los enfrentes, como si los dos se fueran a mirar de frente. Por consola serial debes ir mostrando la lectura de luz de la fotocelda, esta lectura debe ser en porcentaje de 0% a 100%. Un ejemplo de la salida por consola: **Concentración de luz 57%**

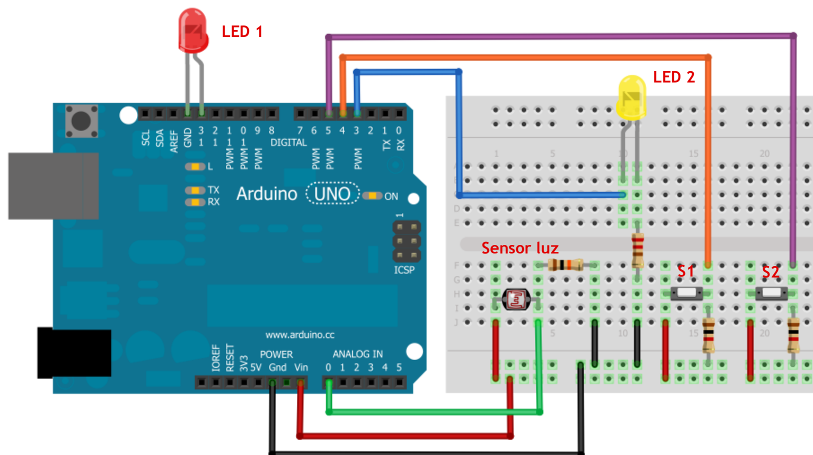


2



La hermosa bióloga **María Paula ;)** es la encargada de cuidar el gran Herbario de Santa Mónica CA donde se preservan diversas plantas. Bilumetium, es un tipo de rosa que requiere a temprana edad iluminación del tipo encendido/apagado, cuando su sensor de luz es mayor a 600 la luz se debe encender (LED1), cuando la planta es adulta se requiere de una iluminación proporcional, en la medida que se va oscureciendo la luz se va encendiendo gradualmente (LED2). Mediante el pulsador S1 se activa el programa de planta joven y mediante el pulsador S2 se activa el programa correspondiente a la planta adulta.





T9 Contador de pulsos

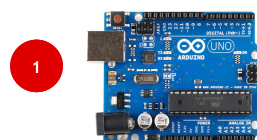
¿Qué aprendo?

- Incrementar una variables
- Condicional If/else anidado
- Anti-rebote de un pulsador
- Leer una entrada digital y escribir una salida digital a determinada condición

Conocimientos previos

- Señal digital
- Función digitalWrite() y digitalRead()
- Imprimir datos por consola
- Declarar variables enteras

Materiales

1 

Arduino UNO

1 

LED Amarillo

1 

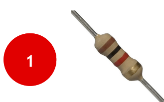
Pulsador

1 

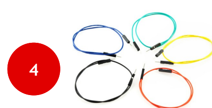
Protoboard

1 

Cable USB Tipo AB

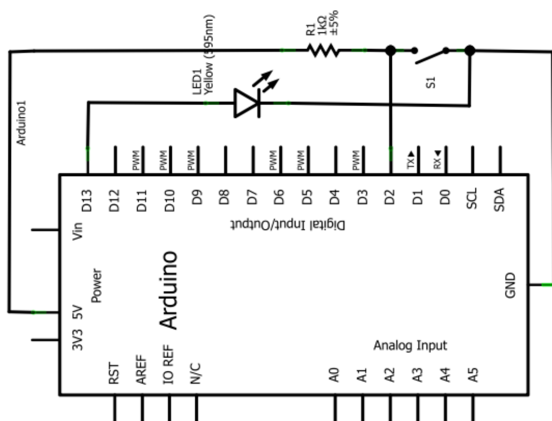
1 

Resistencia 1K

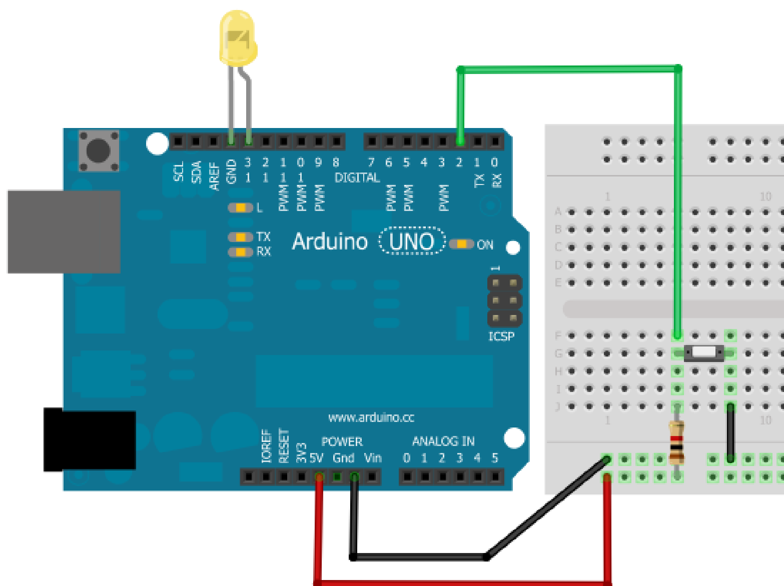
4 

Conectores MM

PLANO

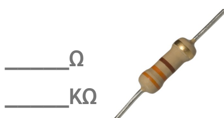


ESQUEMA



PREGUNTA

1— ¿Cuál es el valor de esta resistencia?



______Ω

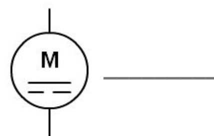
______KΩ

2— ¿Cuáles son los estados de una variable booleana?

a.______________

b.______________

3— Este símbolo a que corresponde



____

CÓDIGO DE PROGRAMACIÓN

```
/*
 -----------------------------------------------
  Contador de pulsos
 -----------------------------------------------

  Programa que muestra por pantalla (consola serial) el número
   de veces que el pulsador ha sido presionado, se realiza un
   proceso que de acuerdo al número de pulsaciones se enciende
   un LED,

 Cosas de Mecatrónica y Tienda de Robótica

*/

//------------------------------------------------------
//Declara puertos de entradas y salidas y variables
//------------------------------------------------------
int conta = 0;  //Variable para guardar el conteo de los pulsos

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  Serial.begin(9600);   //Inicia comunicación serial
  pinMode(2,INPUT);     //Configura el pin 2 como una entrada, pulsador
  pinMode(13,OUTPUT);   //Configura el pin 13 como una salida, LED
}

//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  // Si el pulsador esta oprimido
  if ( digitalRead(2) == HIGH )
  {
      // Si el pulsador no esta oprimido, flanco de bajada
      if ( digitalRead(2) == LOW )
      {
         conta++;               //Incrementa el contador
         Serial.println(conta); //Imprime el valor por consola
         delay (100);           // Retardo
      }
  }

  // Si el valor del contador es 5
  if (conta==5)
  {
    digitalWrite(13,HIGH);  //Enciende el LED
  }

  // Si el valor del contador es 8
  if (conta==8)
  {
    digitalWrite(13,LOW);  // Apaga el LED
  }

}

//Fin programa
```

TIPS

1- Arduino tiene una gran comunidad de aprendizaje y para compartir diversas preguntas, en el foro oficial puedes encontrar diversos temas con gran cantidad de respuestas:

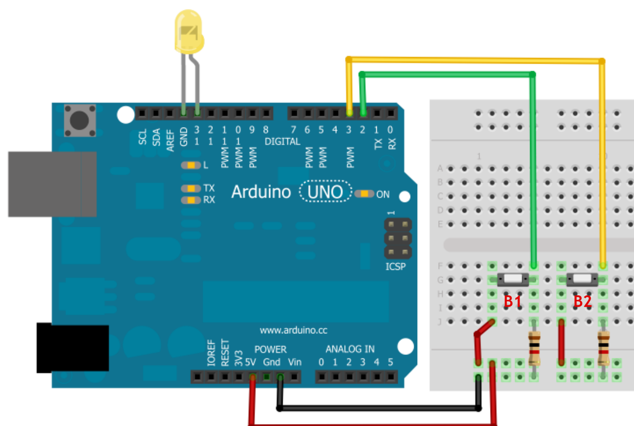
arduino.cc/forum

EJERCICIOS

1

Para la próxima fiesta el DJ Lui Lote te ha contratado para que incorpores un show de luces realmente fantástico, para ello él tiene a disposición un botón 1 (B1) que puede oprimir muchas veces, el número de veces que él pulse es el número de veces que van a prender y apagar las luces a intervalo de 1s. Se utiliza un botón 2 (B2) para confirmar la rutina. Una vez acabe de hacer la rutina el sistema debe quedar listo para darle muchos más pulsos y ver otra rutina distinta.



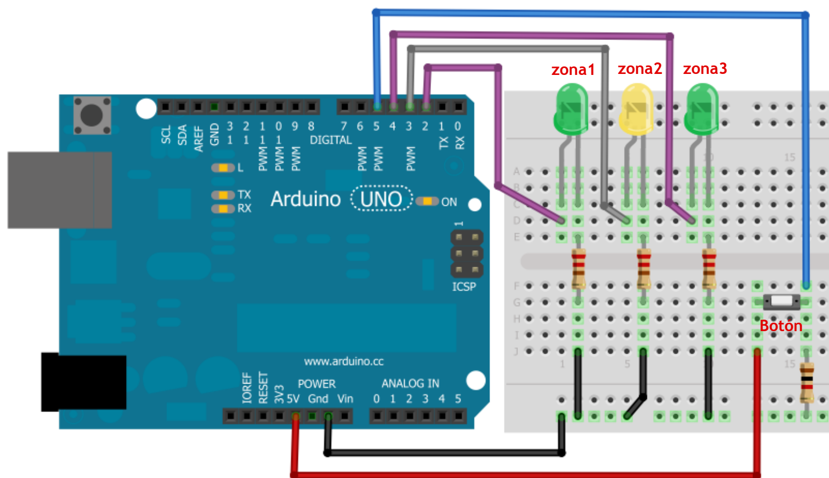


2



El DJ Lui Lote del ejercicio 1 regreso recargado con nuevas ideas para su Show. Ahora cuenta con tres distintas zonas de iluminación para su pista de baile, en su consola de DJ tiene un botón para dar un cierto número de pulsos y luego por la Consola Serial el dice a que zona de iluminación se le asignan esos pulsos a intervalos de 1s. Las palabras que recibe la Consola Serial son: **zona1 zona2 zona3**





T10 Interruptor magnético para una alarma visual

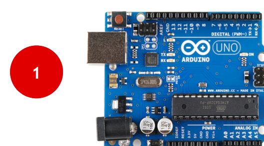
¿Qué aprendo?

- Ciclo for
- Condicional If/else
- Estado de Reed switch
- Leer una entrada digital y escribir una salida digital

Conocimientos previos

- Señal digital
- Función digitalWrite() y digitalRead()
- Parámetros de un for
- Intermitencia de un LED

Materiales

1 

Arduino UNO

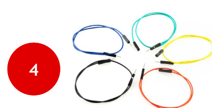
1 
LED Verde

1 
Protoboard

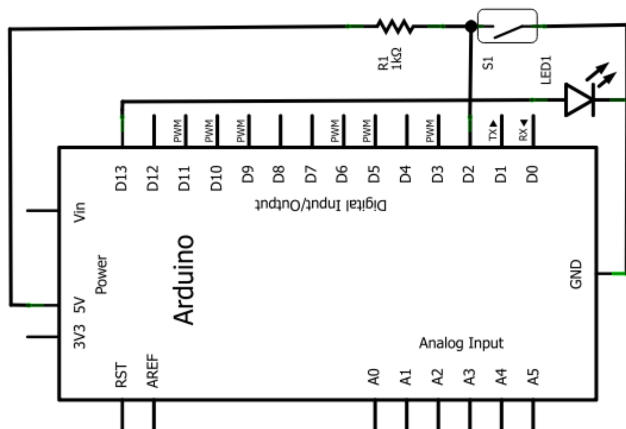
1 
Reed switch

1 
Cable USB Tipo AB

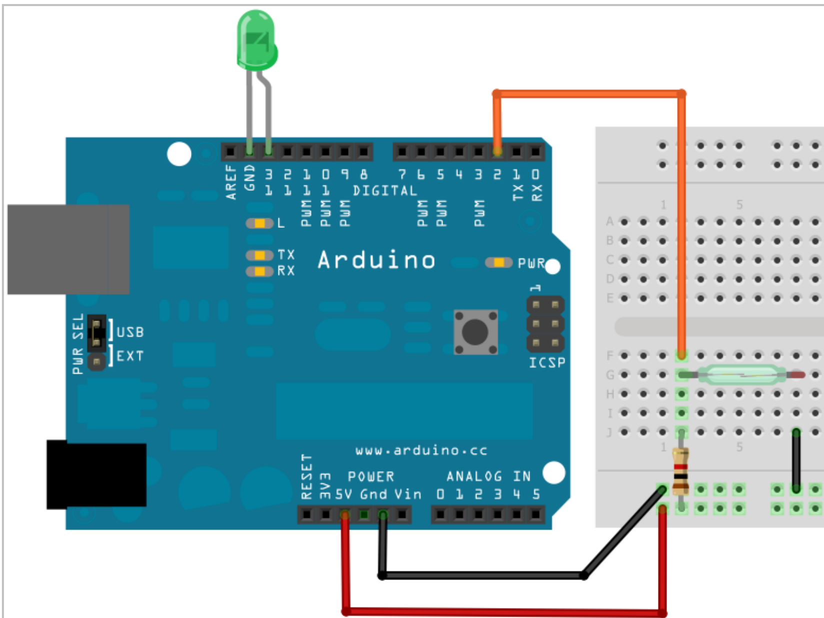
1 
Resistencia 1K

4 
Conectores MM

PLANO

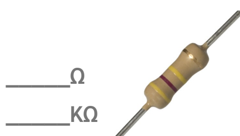


ESQUEMA



PREGUNTA

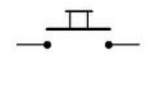
1— ¿Cuál es el valor de esta resistencia?



2— ¿Qué recibe la función **delay()**?

____

3— Este símbolo a que corresponde



____

CÓDIGO DE PROGRAMACIÓN

```
/*
 --------------------------------------------
  Interruptor magnético - Ladrón en casa
 --------------------------------------------

  Programa que genera una alarma visual con un LED
    a partir de un campo magnético generado a un
    reed switch.

 Cosas de Mecatrónica y Tienda de Robótica

*/

//--------------------------------------------------
//Declara puertos de entradas y salidas y variables
//--------------------------------------------------

int contacto = 2;  //Pin asignado al reed switch
int led= 13;       //Pin asignado al LED

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  pinMode(contacto,INPUT); //El reed switch como una entrada
  pinMode(led, OUTPUT);    //El LED como una salida

}

//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  // Si el iman se acerca al reed switch
  if (digitalRead(contacto)==LOW){

     //Ciclo for que va de 0 a 50, el contador esta
     //  en la variable a, se repite mientras a sea
     //  menor a 50
      for(int a=0; a<50; a++){

        digitalWrite(led,HIGH); //Prende el LED
        delay(50); //Tiempo
        digitalWrite(led,LOW); //Apaga el LED
        delay(50); //Tiempo

      }

  // Si el iman esta lejos del reed switch
  }else{

      digitalWrite(led,LOW); //Mantiene apagado el LED

  } //Fin del if


}

//Fin del programa
```

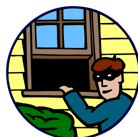
TIPS

1- Estos dispositivos te permitirán desarrollar aplicaciones orientadas a la seguridad de un recinto, al estilo de una alarma antirrobos o sistemas de reconocimiento por parámetros biométricos (huella, iris, voz o rostro)



Cámara a color

Captura a una resolución VGA



Sensor de vibración

Detecta vibraciones en superficies horizontales





PIR

Sensor para detectar movimiento

Micrófono MEMS

Soporta un alto rango de ruido



EJERCICIOS

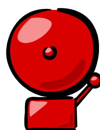
1



Una casa requiere un sistema de alarma, para ello en este ejercicio te proponemos que emulemos una. Cuando todos salen de la casa la última persona en salir se encarga de activar la alarma mediante un número de 4 dígitos, en ese momento el sistema comienza a recibir información de los detectores magnéticos (reed switch) ubicados en cada ventana y en cada puerta. La alarma se debe disparar cuando un campo magnético altere el estado del reed switch, cuando la alarma se activa se comienza emitir un sonido y además se genera una alarma visual, la única forma de desactivar la alarma es mediante un código numérico de 4 dígitos que debe ser diferente al código de activación, los pasos se describen a continuación:

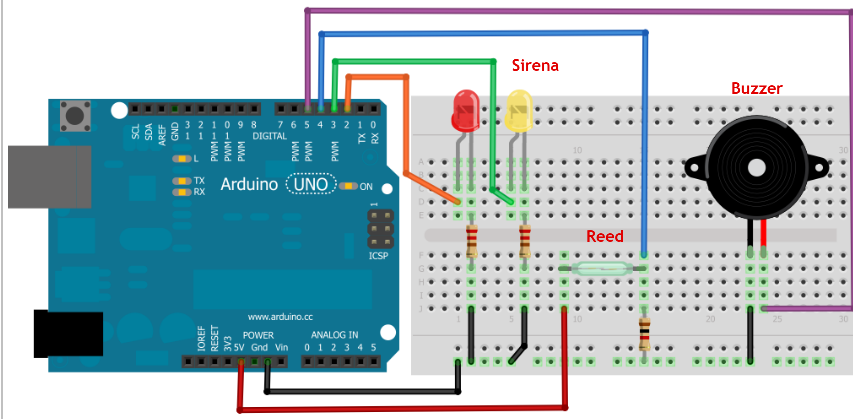
A- Por consola serial se recibe un número entero de 4 dígitos, si este número es igual al guardado por el sistema el sistema de alarma se activa, para ello se debe mostrar por consola "Alarma activada"

B- Si el Reed switch presencia algún campo magnético, la consola serial debe mostrar "Presencia de extraño, sistema disparado"



C- Como la alarma se disparo se debe encender el buzzer para generar sonido (se maneja como una salida digital) y además el LED amarillo y el rojo se debe alternar encendido/apagado y emular los movimientos de una sirena de bomberos a intervalos de 1s.

D- Si la alarma se quiere desactivar se debe ingresar un código numérico entero de 4 dígitos de ser correcto mostrar por consola serial "Alarma desactivada" de lo contrario el paso C se debe repetir mientras no se ingrese el código correcto.



MI CUADERNO DE APUNTES

T11 LED RGB apoyado de tabla de colores

¿Qué aprendo?

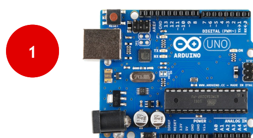
- Manejar el código de colores RGB
- Uso de una función
- PWM a tres salidas
- Manipular una variable

Conocimientos previos

- Señal digital
- Función analogWrite()
- LED de anodo común
- Estructura de un programa en Arduino

Materiales

1



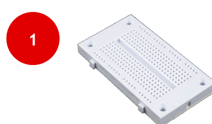
Arduino UNO

1



LED RGB

1



Protoboard

1



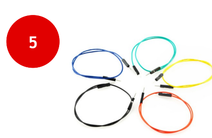
Cable USB Tipo AB

3



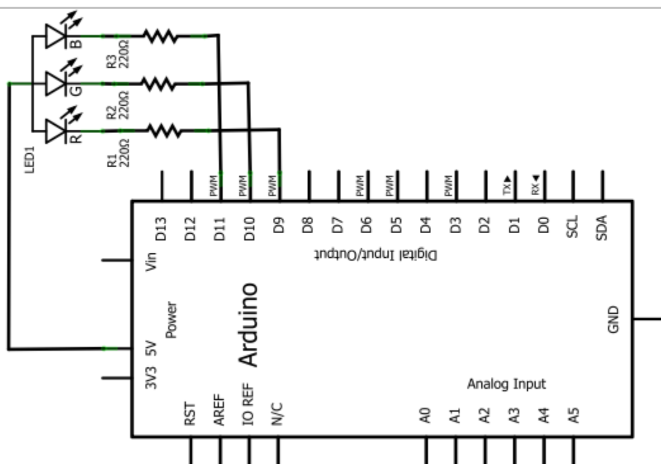
Resistencia 220

5

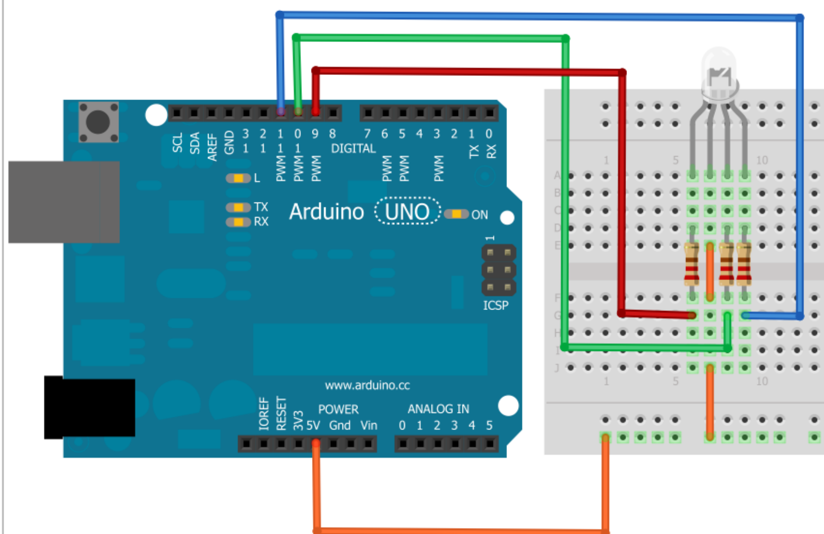


Conectores MM

PLANO



ESQUEMA

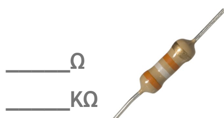


PREGUNTA

1— ¿Cuál es el valor de esta resistencia?

______Ω

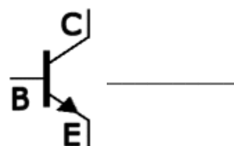
______KΩ



2— La palabra **"pulso"** se debe guarda en una variable de tipo :

____

3— Este símbolo a que corresponde



____

CÓDIGO DE PROGRAMACIÓN

```
/*
 -----------------------------------------------
  LED RGB - Tabla de Colores
 -----------------------------------------------

  Programa que hace uso de una función llamada
   color para generar diversas tonalidades en
   un LED RGB

 Cosas de Mecatrónica y Tienda de Robótica

*/

//-----------------------------------------------------
//Declara puertos de entradas y salidas y variables
//-----------------------------------------------------
int ledRojo = 9; //Declara Pin LED Rojo
int ledVerde=10; //Declara Pin LED Verde
int ledAzul=11;  //Declara Pin LED Azul

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  pinMode(ledRojo,OUTPUT);  //El LED Rojo como una salida
  pinMode(ledVerde,OUTPUT); //El LED Verde como una salida
  pinMode(ledAzul,OUTPUT);  //El LED Azul como una salida
}

//------------------------------------
//Funcion ciclica
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

   // Llamado a la función Color que recibe
   // 1er posicion: Intensidad Rojo
   // 2da posición: Intensidad Verde
   // 3ra posición: Intensidad Azul

   color(255,0,0);     //Rojo
   delay(1000);

   color(0,255,0);     //Verde
   delay(1000);

   color(0,0,255);     //Azul
   delay(1000);

   color(255,255,255); //Blanco
   delay(1000);

   color(255,0,255);   //Magenta
   delay(1000);

   color(255,128,0);  //Naranja
   delay(1000);

}

//------------------------------------
//Funcion color
//------------------------------------

void color(int rojo, int verde, int azul){

    //Escritura de PWM del color Rojo
    analogWrite(ledRojo, 255-rojo);

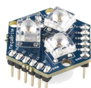
    //Escritura de PWM del color Verde
    analogWrite(ledVerde, 255-verde);

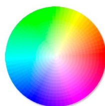
    //Escritura de PWM del color Azul
    analogWrite(ledAzul, 255-azul);
}

//Fin programa
```

TIPS

1- Estos dos dispositivos pueden generar hasta un billón de colores distintos


MegaBrite



ShiftBrite


EJERCICIOS

1



La iluminación de la recepción del Hotel Mancherie es algo especial porque de acuerdo al estado de animo del dueño el Sr. Martin, se debe configurar la luz en una escala RGB. El operario de las luces es muy pilo ya que se encarga de recibir los colores deseados por el Sr. Martin, así que el operario se encarga de escribir por Consola Serial un valor RGB separado por comas (,). Así que debes desarrollar un programa que tome esta información y la aplique a un LED RGB, toma como Esquema de la pág 103. Por ejemplo si se envía:

255,0,0 -> LED RGB muestra el tono Rojo
153,255,0 -> LED RGB muestra un Verde limón
0,102,255 -> LED RGB muestra un Azul cielo



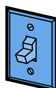


2

El operario del punto anterior se va de vacaciones por lo cual el Sr. Martin esta muy molesto porque no entiende muy bien como escribir el color separado por comas. Por ello al operario se le ha ocurrido una gran idea, colocar dos pulsadores:

Pulsador 1 (S1) permite escoger un color dentro de un menú (Rojo, Verde o Azul), cada vez que se pulse S1 el texto por Consola Serial que dice el color debe cambiar, debe tener antirebote para que por cada pulso dado solo cambie un color. Un ejemplo del pulso dado y el valor mostrado por consola:

Pulso 1 -> Rojo
Pulso 2 -> Verde
Pulso 3 -> Azul
Pulso 4 -> Rojo

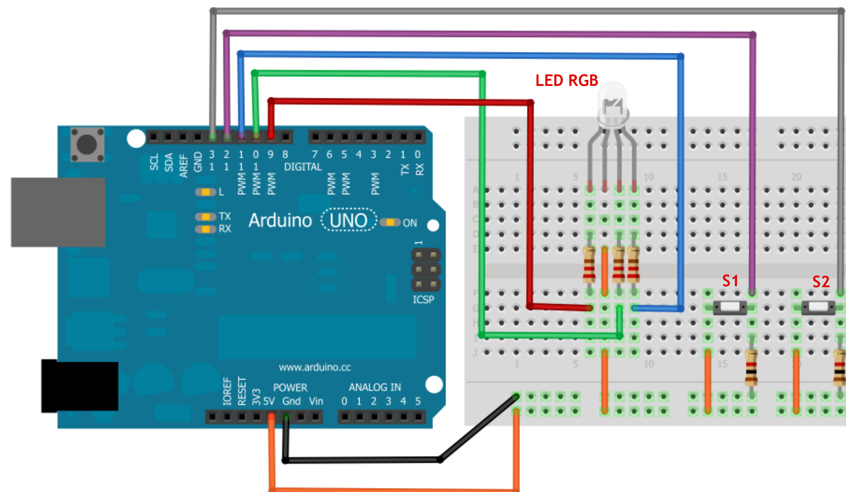


Pulsador 2 (S2) permite variar el color de 0 a 255 del último nombre del color que quedo asignado mediante el Pulsador 1 (S1). Mientras se mantenga pulsado S2 a intervalos de 100ms (milisegundos) la variable de ese color se debe ir incrementando de uno en uno. Se debe imprimir el nombre del Color (último estado debido a S1) y a continuación el valor numérico del color, finalmente el resultado se debe ir viendo reflejado en el LED RGB. Por ejemplo de acuerdo al último estado del Pulso 4 y luego se pulsa S2:

Pulso 4 -> Rojo
Rojo 0
100ms
Rojo 1
100ms
Rojo 2



Ayudas para resolverlo: Por cada color debes crear dos variables, una para guardar el nombre del color y la otra para guardar el valor numérico que llevas de ese color.



T12 Control ON/OFF de un motor

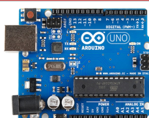
¿Qué aprendo?

- Control ON/OFF
- Condicional If/else
- Conectar un motor DC por transistor
- Condicionales a partir del estado del pulsador

Conocimientos previos

- Señal digital
- Función digitalWrite() y digitalRead()
- Divisor de voltaje
- Reconocer un transistor y un motor

Materiales

1 

Arduino UNO

1 

Transistor NPN

1 

Protoboard

1 

Cable USB Tipo AB

1 

Pulsador

1 

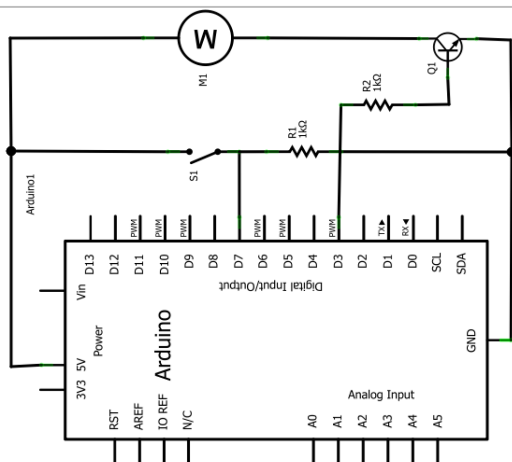
Motor DC

2 

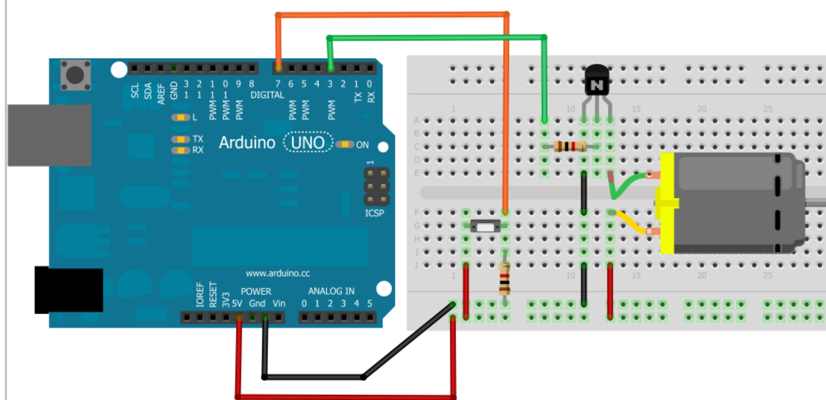
Resistencia 1K

8 

Conectores MM



PLANO

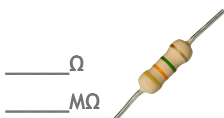


ESQUEMA

1— ¿Cuál es el valor de esta resistencia?

______Ω

______MΩ



2— Si quiero guardar el número **π** que tipo de variable debo usar?

____

3— Este símbolo a que corresponde



____

PREGUNTA

CÓDIGO DE PROGRAMACIÓN

```
/*
 -------------------------------------------
  Control ON/OFF de un motor
 -------------------------------------------

  Programa que hace uso de un motor y un pulsador,
   mientras se mantenga pulsado, el motor debe
   estar encendido (ON) de lo contrario debe estar
   apagado (OFF)

 Cosas de Mecatrónica y Tienda de Robótica

*/


//--------------------------------------------------
//Declara puertos de entradas y salidas y variables
//--------------------------------------------------
int pulsador =7; //Declara Pin del pusaldor
int motor=3;     //Declara Pin del motor


//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  pinMode(pulsador,INPUT); //El pulsador como una entrada
  pinMode(motor,OUTPUT);   //El motor como una salida
}


//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino


  // Si el pulsador se encuentra oprimido
  if(digitalRead(pulsador) == HIGH){

    digitalWrite(motor,HIGH); //Enciende el motor

  }else{ //si el pulsador no esta oprimido

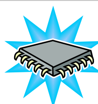
    digitalWrite(motor,LOW); //Apaga el motor

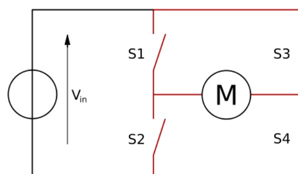
  }


}
// Fin programa
```

TIPS

1- Un **Puente H** es un circuito electrónico que permite a un motor DC girar en ambos sentidos, *avance* y *retroceso*. Son ampliamente usados en robótica y como convertidores de potencia.



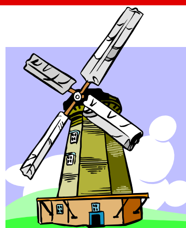


Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos. El término "puente H" proviene de la típica representación gráfica del circuito. Un puente H se construye con 4 interruptores (mecánicos o mediante transistores). Cuando los interruptores S1 y S4 están cerrados y S2 y S3 abiertos se aplica voltaje positivo en el motor, haciéndolo girar en un sentido. Abriendo los interruptores S1 y S4 y cerrando S2 y S3, el voltaje se invierte, permitiendo el giro en sentido inverso del motor. Con la nomenclatura que estamos usando, los interruptores S1 y S2 nunca podrán estar cerrados al mismo tiempo, porque esto cortocircuitaría la fuente de tensión. Lo mismo sucede con S3 y S4.



EJERCICIOS

1 Un molino para extraer agua subterránea tiene un motor que ayuda al proceso de bombeo del líquido, el agua se encuentra a una profundidad de 40m y la **motobomba** debe funcionar de manera continua. El granjero cuenta con un único pulsador (S1), al oprimir este por primera vez se enciende la motobomba si el vuelve a oprimir el mismo pulsador por segunda vez la motobomba se apaga con lo cual se deja de bombear el líquido. El montaje de este ejercicio usa el mismo Esquema de conexiones que el expuesto en la página 107.



2

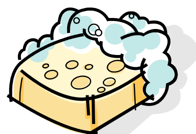


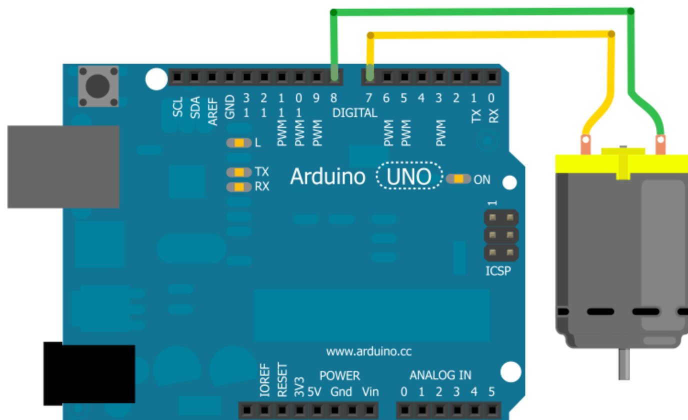
Al mercado ha salido un nuevo detergente para lavadoras, con una gran novedad, si cuando uses el deterge en tu lavadora esta bate hacia la derecha **(D)** a determinado tiempo es posible remover las manchas en la ropa mientras que si la lavadora bate a la izquierda **(I)** a determinado tiempo al final del ciclo de lavado la ropa tendrá una textura muy sedosa como si se le hubiera aplicado suavizante. La empresa **Lucky Luc** gran fabricante de lavadores quiere implementar este sistema, para ello te ha contratado para que desarrolles el programa que de acuerdo a lo que se reciba por consola se debe operar la lavadora, por ejemplo:

D,3 Lavadora bate a la derecha por 3 segundos

I,2 Lavadora bate a la izquierda por 2 segundos

A Lavadora apagada





MI CUADERNO DE APUNTES

T13 Control por PWM de un motor

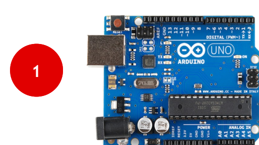
¿Qué aprendo?

- Leer datos de la Consola Serial
- Manejo de la función map()
- Variar el PWM para producir 5 velocidades distintas
- Etapa de potencia para un motor a través de transistor

Conocimientos previos

- Señal análoga
- Función analogWrite() y Serial.print()
- PWM
- Condicional y operadores de comparación

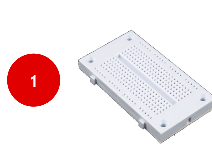
Materiales

1 

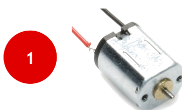
Arduino UNO

1 

Transistor NPN

1 

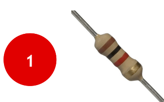
Protoboard

1 

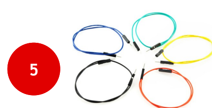
Motor DC

1 

Cable USB Tipo AB

1 

Resistencia 1K

5 

Conectores MM

PLANO

ESQUEMA

PREGUNTA

1— ¿Cuál es el valor de esta resistencia?

______Ω

______KΩ

2— ¿Cómo se escriben los siguientes números en binario?

7=______

5=______

2=______

3— Este símbolo a que corresponde

____

CÓDIGO DE PROGRAMACIÓN

```
/*
 ---------------------------------------------
  Control por PWM de un motor
 ---------------------------------------------

  Programa que hace uso de un motor y la Consola
   serial de Arduino, tiene la posiblidad de
   configurar al motor 5 velocidades distintas,
   desde el teclado del PC puedes enviarle la
   velocidad deseada. Las 5 velocidades se configuran
   con 5 PWM distintos.

 Cosas de Mecatrónica y Tienda de Robótica

*/

//--------------------------------------------------
//Declara puertos de entradas y salidas y variables
//--------------------------------------------------
int motor=3;     //Declara Pin del motor

//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  Serial.begin(9600); //Inicia la comunicacion serial Arduino-PC
}

//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  // Si hay algun valor en la Consola Serial
  if (Serial.available()){

      //Variable donde se guarda el caracter enviado desde teclado
      char a = Serial.read();

      // Si el caracter ingresado esta entre 0 y 5
      if (a>='0' && a<='5'){

        //Variable para escalar el valor ingresado a rango de PWM
        int velocidad = map(a,'0','5',0,255);
        //Escritura de PWM al motor
        analogWrite(motor,velocidad);
        //Mensaje para el usuario
        Serial.print("El motor esta girando a la velocidad ");
        Serial.println(a);

      }else{ // Si el caracter ingresado NO esta entre 0 y 5

        //Mensaje para el usuario
        Serial.print("Velocidad invalida");
        Serial.println(a);

      }

  }

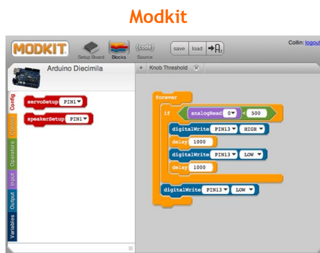
}

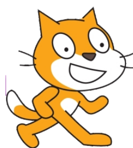
//Fin programa
```

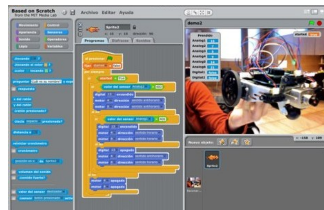
TIPS

1- Arduino también se puede programar en lenguajes gráficos, por ejemplo:

Modkit







Scratch for Arduino

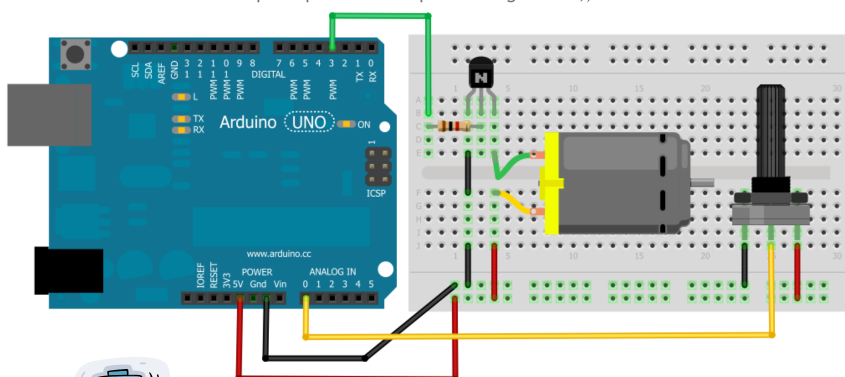
EJERCICIOS

1





El director de la película **Súper Cocodrilo** debe grabar una escena donde Súper Cocodrilo recibe fuertes vientos para hacer que su capa de súper héroe se mueva al unísono a la velocidad del viento, para ello tiene un gran ventilador (motor) de alta potencia, el control de la velocidad del ventilador la tiene el director, en la medida que él gire el potenciómetro la velocidad debe ir cambiando. Tu misión es desarrollar un programa que cumpla con estas características para que la escena pueda ser grabada ;)



2



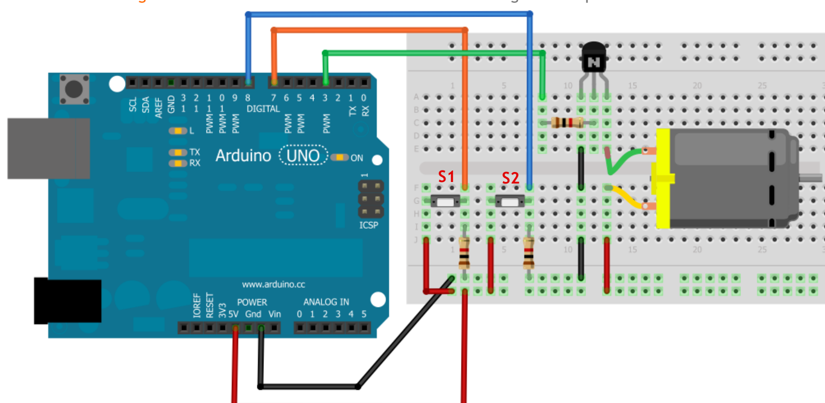
La última novedad de la Feria de Electrodomésticos ha dejado con gran renombre una licuadora que incorpora dos pulsadores mediante los cuales es posible variar las 10 diferentes velocidades (0 a 9) al momento de licuar. Todo esto se apoya desde la Consola serial donde es posible ver la Velocidad escogida, los pulsadores se ajustan de la siguiente manera:

Pulsador 1 (S1) permite escoger una velocidad dentro de un menú (Velocidad 0 ... Velocidad 9), cada vez que se pulse S1 el texto por Consola Serial que dice la velocidad debe cambiar, debe tener antirebote para que por cada pulso dado solo cambie una velocidad. Un ejemplo del pulso dado y el valor mostrado por consola:

Pulso 1 -> Velocidad 0
Pulso 2 -> Velocidad 1
Pulso 3 -> Velocidad 2



Pulsador 2 (S2) al pulsar S2 por primera vez el motor debe comenzar a girar a la velocidad escogida mediante S1, si se pulsa por segunda vez S2 el motor debe parar. Por Consola serial se debe mostrar "Licuadora girando a la velocidad x" o "Licuadora detenida" según corresponda.



T14 Generar tonos con un buzzer

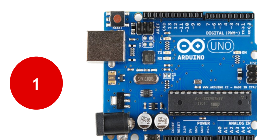
¿Qué aprendo?

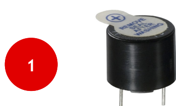
- Manejo de variables de tipo entera
- Usar funciones especiales de Arduino
- Generar diversos tonos
- Producir salidas de frecuencia

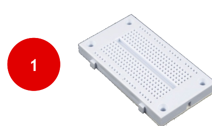
Conocimientos previos

- Señal digital y análoga
- Función map() y analogRead()
- Enviar parámetros a las funciones
- Retardos a través de delay()

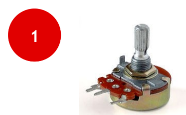
Materiales

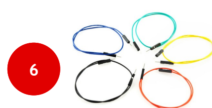
1 
Arduino UNO

1 
Buzzer

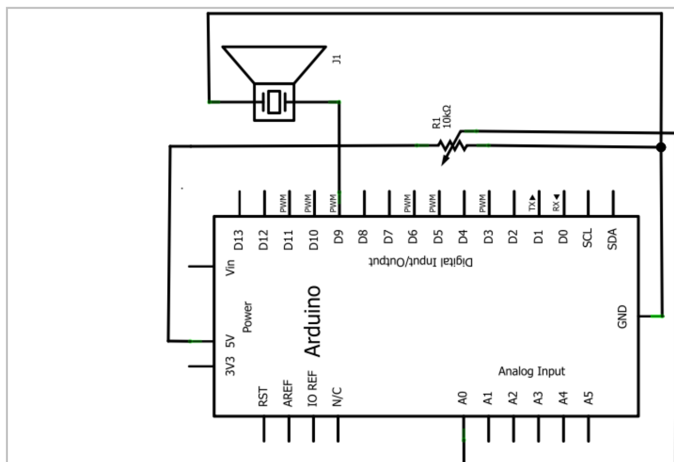
1 
Protoboard

1 
Cable USB Tipo AB

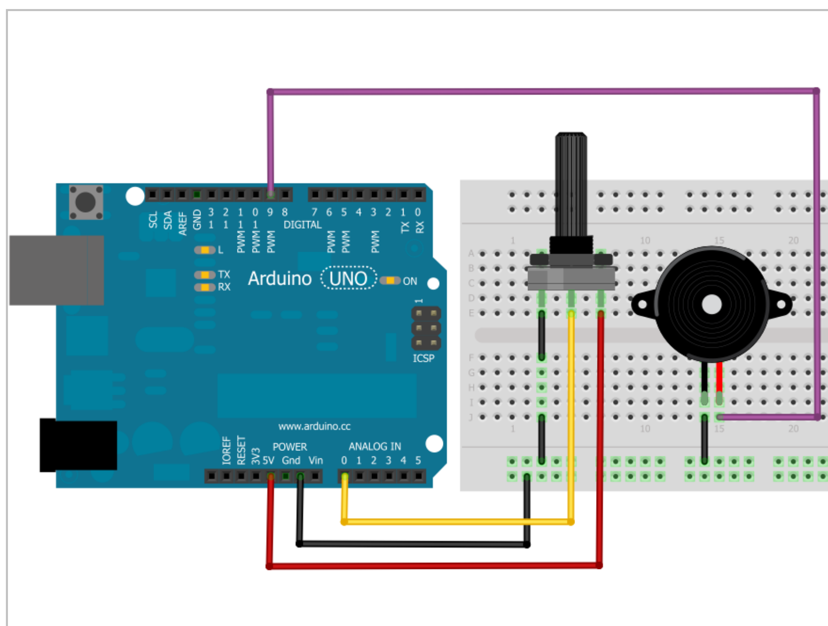
1 
Potenciómetro 10K

6 
Conectores MM

PLANO

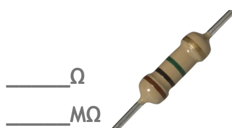


ESQUEMA



PREGUNTA

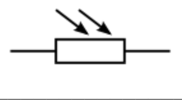
1— ¿Cuál es el valor de esta resistencia?



2— El **switch...case** es un tipo de

____

3— Este símbolo a qué corresponde



____

CÓDIGO DE PROGRAMACIÓN

```
/*
 -----------------------------------------------
 Generar tonos con un buzzer
 -----------------------------------------------

 Programa que hace uso de un buzzer (chicharra) y un
 potenciómetro, la idea es generar diversos tonos
 en el buzzer a partir del estado análogo del
 potenciómetro. Además se hace uso de la función
 tone que es muy útil para generar diversas melodías

 Cosas de Mecatrónica y Tienda de Robótica
*/


//------------------------------------------------------
//Declara puertos de entradas y salidas y variables
//------------------------------------------------------
int buzzer = 9;  //Declara Pin del buzzer
int tono = 0;    //Declara Pin del potenciómetro


//------------------------------------
//Funcion principal
//------------------------------------
void setup() // Se ejecuta cada vez que el Arduino se inicia
{
  // No se configuran parámetros inciales, pero se debe
  //  colocar el encabezado de la función setup()
}


//------------------------------------
//Funcion ciclicla
//------------------------------------
void loop() // Esta funcion se mantiene ejecutando
{           //  cuando este energizado el Arduino

  //Variable entera donde se almacena el valor del potenciómetro
  int sensor = analogRead(tono);

  //Variable donde se escala la frecuencia de 100 a 5000Hz
  int frecuencia = map(sensor,0,1023,100,5000);

  //Variable entera para guardar el tiempo deseado en ms
  int duracion = 250;

  //Funcion tone(), que recibe:
  //  1ra posición: Pin del elemento sonoro
  //  2da posición: Frecuencia deseada en Hz
  //  3ra posición: Duración del tono
  tone(buzzer, frecuencia, duracion);

  //Retardo
  delay(100);
}

//Fin programa
```

TIPS

1- Si tienes dificultades para aprender los colores de las resistencias una ayudita online no te caería nada mal ;) en **Wolfram Alpha** puedes hacer esto:

WolframAlpha computational knowledge engine

resistor yellow blue brown

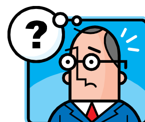
Examples Random

Input interpretation:
resistor | yellow | blue | brown

Resistance:
0.46 kΩ (kilohms)

Image: Show chart

Unit conversions:
460 Ω (ohms)

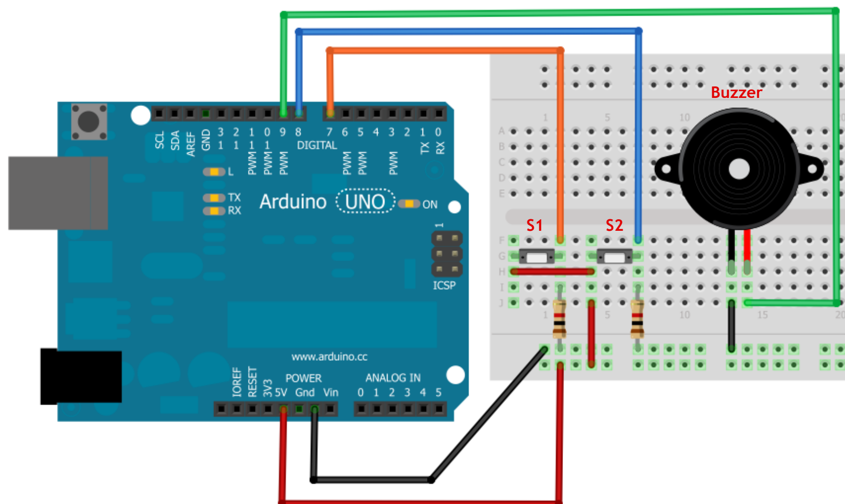


EJERCICIOS

1



Te proponemos hacer un sintetizador musical de dos tonos, usando dos pulsadores. La idea es que al pulsar S1 se produzca un tono a una frecuencia de 400Hz con una duración de 100ms, mientras si presionas S2 se produce un tono a frecuencia de 2600Hz con una duración de 200ms. Combina estos dos tonos hasta que desarrolles una bonita melodía.



MI CUADERNO DE APUNTES



A
P
O
R
T
E
S

Hola, queremos conocer tus opiniones referente a este material, son de gran ayuda con el ánimo de seguir mejorando los contenidos y haciendo éstos más claros.



Estamos atentos a recibir todo tipo de comentarios que nos sirvan de realimentación y fortalezcan más nuestra comunidad de conocimientos.

Guía básica de Arduino

Hazlo tú mismo y aprende electrónica y programación

- **9 Capítulos de contenido**
- **14 Tutoriales paso a paso**
- **28 Ejercicios aplicados a casos de la vida real**
- **Preguntas, tips, quices y mucho más**



Contáctenos

tiendaderobotica.com

ventas@tiendaderobotica.com



tdrobotica


tdrobotica



CosasdeMecatronica